

DTIC FILE COPY

2



# United States Army Recruiting Command

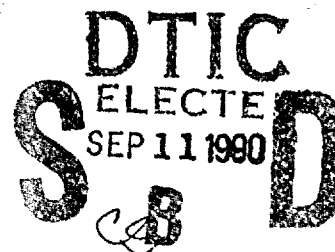
---

## USAREC SR 89-9

---

### RECRUITER'S APPLICANT PROCESSING SYSTEM (RAPS)

BY



DR ANDRES INN  
SANDY ADAMS  
AND  
GENE BELLINGER

OCTOBER 1989

90 09 10 260

APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION UNLIMITED

---

Research & Studies Division  
U.S. Army Recruiting Command  
Program Analysis and Evaluation Directorate  
Fort Sheridan, Illinois 60037-6090

AD-A226 417

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 20 October 1989		3. REPORT TYPE AND DATES COVERED	
4. TITLE AND SUBTITLE Recruiter's Applicant Processing System (RAPS) Final Report				5. FUNDING NUMBERS DAKF15-87-D-0144 SubH188-08 D.O. 008	
6. AUTHOR(S) Andres Inn Sandra Adams GeneBellinger					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) I.C.S. — Suite 505-S Crystal Towers 1600 S. Eads Street Arlington, VA 22202 <i>HumRRO International Inc Alex VA</i>				8. PERFORMING ORGANIZATION REPORT NUMBER <i>549-2611</i>	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Recruiting Command USARCRO-S Fort Sheridan, IL 60037-6000				10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION/AVAILABILITY STATEMENT				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) <p>→ RAPS represents a proof-of-concept in the development of a user-friendly expert system designed to be a recruiter's assistant. RAPS is designed as a full-featured system. It codifies the Army 601-210 regulations, it functions as a recruit qualifier and a form builder. By enabling multiple functions, including simplifying the maintenance of Lear Response Lists, REACT files, 200 cards, etc., RAPS is expected to be viewed as a valuable recruiter's assistance.</p> <p><i>user interface, man, systems engineering</i>      <i>Keywords:</i></p>					
14. SUBJECT TERMS Expert System, Artificial Intelligence, Recruiting, Database. <i>(KRE)</i>				15. NUMBER OF PAGES 163	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT UL		

**Recruiter's Applicant Processing System  
(RAPS)  
Final Report**

Prepared for:

U.S. Army Recruiting Command  
Ft. Sheridan, IL

Prepared by:

Dr. Andres Inn  
Sandy Adams  
Gene Bellinger

I.C.S.

October 20, 1989

## Table of Contents

1. Introduction	1
2. Initial Design Concepts	3
2.1. Design Overview	3
2.1.1. The Recruit Qualifier	4
2.1.2. The Form Builder	5
3. Insights / Concerns of Individuals and Organizations	8
3.1. Applicant	8
3.2. Recruiter	9
3.3. Recruiting Station	11
3.4. MEPS and METS	12
3.5. USAREC	13
3.6. Department of the Army	14
3.7. Summary	14
4. General RAPS Features	15
4.1. Windows	15
4.2. Menus	15
4.3. Dialog boxes	16
4.4. Dialog box Elements	17
4.4.1. Choice Buttons	17
4.4.2. Radio Buttons	17
4.4.3. Edit Field	18
4.4.4. Push Button	18
4.4.5. List Box	18
4.4.6. Edit Box	18
4.4.7. Text	18
4.5. Moving Through and Selecting from Dialog Boxes	19
4.5.1. Choice Button	19
4.5.2. Radio Button	19
4.5.3. Edit Field	19
4.5.4. Push Button	20
4.5.5. List Boxes	20
4.5.6. Edit Box	21
4.5.7. Hot-Keys	22
4.5.8. Exiting a Dialog box	22
5. RAPS Design / Operation	23
5.1. RAPS Windows	23
5.1.1. RAPS Menu Bar and Pull-Down Menus	26
5.1.2. Menu Greying	27
5.1.3. Menu Checking	27
5.1.4. RAPS Help	28
5.1.5. RAPS Status Bar	30



Codes

Avail and/or  
Special

Dist

A-1

5.1.6.	RAPS Date	30
5.2.	Recruiter Access Phase	31
5.2.1.	Recruiter Menu	31
5.2.2.	Recruiter Login Dialog	33
5.2.3.	Register Recruiter Dialog	33
5.2.4.	Change Login Dialog	34
5.2.5.	Delete Recruiter Dialog	35
5.2.6.	Quit Dialog	36
5.2.7.	Save Database Dialog	37
5.3.	Lead-Generation Phase	38
5.3.1.	Select Lead Dialog	38
5.3.2.	Update Lead Dialog	39
5.3.3.	Add Lead Dialog	40
5.3.4.	Import Lead Dialog	41
5.3.5.	Delete Lead(s) Dialog	42
5.3.6.	Print Lead(s) Dialog	43
5.4.	Prospecting Phase	45
5.4.1.	Select Lead Dialog	46
5.4.2.	Update Lead Priority Dialog	47
5.4.3.	Update Lead Contacts Dialog	48
5.4.4.	Update Lead Disposition Dialog	50
5.4.5.	Delete Lead(s) Dialog	51
5.4.6.	Print Lead(s) Dialog	52
5.5.	Selling Phase	54
5.5.1.	Select Prospect Dialog	55
5.5.2.	Update Prospect Data Dialog	55
5.5.3.	Update Prospect Prequalification Data Dialog	57
5.5.4.	Update Prospect Citizenship Data Dialog	59
5.5.5.	Update Prior Service Data Dialog	60
5.5.6.	Update Law Violations Data Dialog	61
5.5.7.	Update Personal Data Dialog	62
5.5.8.	Test Data Dialog	63
5.5.9.	Test Data/Physical Dialog	64
5.5.10.	Interview Remarks Dialog	65
5.5.11.	Disposition Dialog	66
5.5.12.	Delete Prospect(s) Dialog	67
5.5.13.	Print Prospect(s) Dialog	68
5.5.14.	MS-DOS Access	69
5.6.	Processing Phase	71
5.6.1.	Select Applicant Dialog	72
5.6.2.	Update Education Data Dialog(s)	73
5.6.3.	Update Residence Data Dialog	74
5.6.4.	Update Employment Data Dialog	76

5.6.5.	Update Relations Data Dialog	78
5.6.6.	Update Law Violations Data Dialog	79
5.6.7.	Form 714 Request for Examination Update Dialog	81
5.6.8.	Transmit Form 714 Dialog	81
5.6.9.	Disposition Dialog	82
5.6.10.	Delete Applicant(s) Dialog	83
5.6.11.	Print Applicant(s) Dialog	84
5.7.	Enlistment Maintenance Phase	86
5.7.1.	Select DEP/DTP Dialog	86
5.7.2.	Update DEP/DTP Information	87
5.7.3.	Update DEP/DTP Function Data	88
5.7.4.	Update DEP/DTP Law Violations	89
5.7.5.	Update DEP/DTP Loss Data	91
5.7.6.	Disposition Dialog	91
5.7.7.	Delete DEP/DTP(s)	92
5.7.8.	Print DEP/DTP(s) Dialog	94
6.	RAPS Knowledge Base	96
6.1	Overview	96
6.2	Knowledge Base Maintenance	97
6.3.	Knowledge Base Rule Format	100
6.4.	Rules and Inference Engine Control	101
6.5.	The RAPS Format Rule	103
6.6.	RAPS Inference Engine Interface	104
7.	RAPS Software Organization	107
7.1.	RAPS Compiled Files	107
7.1.1.	RAPS.ARI - Raps Main Program	107
7.1.2.	RAPSDCV.ARI - Dialog Control Variables	108
7.1.3.	RAPSDHF.ARI - Dialog Handler Functions	108
7.1.4.	RAPSDHS.ARI - Dialog Handler Support Functions	108
7.1.5.	RAPSKBI.ARI - Knowledge Base Inference Engine	108
7.1.6.	RAPSRSD.ARI - Record Sequence Definitions	108
7.1.7.	RAPSUTL.ARI - Utility Predicates File	109
7.2.	RAPS Run Time Files	109
7.2.1.	RAPSDDC.ARI - Default Dialog Choices (run time)	109
7.2.2.	RAPSDLG.ARI - Dialog Definitions (run time)	109
7.2.3.	RAPSDLG.HLP - Dialog Help File (run time)	109
7.2.4.	RAPSKBR.ARI - Knowledge Base Rules (run time)	110
7.2.5.	RAPSMNU.ARI - Menu Definition (run time)	110
7.2.6.	RAPSMNU.HLP - Menu Help File (run time)	110
7.3.	Running RAPS from the Arity Interpreter	110
7.4.	Running Compiled RAPS from DOS	111
7.5.	RAPS Database Structure	111
7.6.	RAPS Program Architecture	112

7.6.1. RAPS Global Variables and Predicates	113
8. RAPS Printer Interface	136
8.1. RAPS Printer Predicates	137
9. CONCLUSION	141
9.1. The RAPS Recruit Qualifier	141
9.2. The RAPS Form Builder	142
9.3. RAPS Tuning and Extensions	142
9.3.1. RAPS User Interface	142
9.3.1.1. Dialog Box Tuning	143
9.3.1.2. Inference Engine Access and Control	143
9.3.2. RAPS Extensions	144
9.3.2.1. Forms Printing	144
9.3.2.2. Field Format Validation	145
9.3.2.3. Data Communications	145
9.3.2.4. Record Length Optimization	145
9.3.2.5. Help Files	145
9.3.3. Additional RAPS Features	146
9.3.3.1. RAPS Access by Company and Battalion Leadership	146
10. RECOMMENDATIONS	148
10.1. User-Developer Testing and Tuning	148
10.2. Rule Construction and Inference Engine Adjustment	148
10.3. Point-and-Shoot Support	149
10.4. Data and Text Compression Algorithms	149
10.5. Forms Generation and Development	150
10.6. MANDEX Implementation	150
10.7. Communications module	150
10.8. Review of USAREC procedures, operations, and data requirements	151
10.9. RAPS Deployment	151
Appendix A - Glossary	153

## Table of Figures

Figure 5. RAPS Design Concept	24
Figure 5.1. RAPS Windows	25
Figure 5.1.1(a). RAPS Menu Bar	26
Figure 5.1.1(b). RAPS Pull-Down Menu	27
Figure 5.1.3. RAPS Menu Checking	28
Figure 5.1.4(a). RAPS Help	29
Figure 5.1.4(b). RAPS Find Help	30
Figure 5.1.5. RAPS Status Bar	30
Figure 5.2.1. Recruiter Pull-Down Menu	31
Figure 5.2.2. Recruiter Login Dialog	33
Figure 5.2.3. Register Recruiter Dialog	34
Figure 5.2.4. Change Login Dialog	35
Figure 5.2.5. Delete Recruiter Dialog	36
Figure 5.2.6(a). Quit Dialog	37
Figure 5.2.6(b). Save Database Dialog	37
Figure 5.3. Lead Pull-Down Menu	38
Figure 5.3.1. Select Lead Dialog	39
Figure 5.3.2. Update Lead(s) Dialog	40
Figure 5.3.3. Add Lead(s) Dialog	41
Figure 5.3.4. Import Leads Dialog	42
Figure 5.3.5. Delete Lead(s) Dialog	43
Figure 5.3.6. Print Lead(s) Dialog	44
Figure 5.4. Prospecting Menu	45
Figure 5.4.1(a). Select Lead Dialog	46
Figure 5.4.1(b). Telephone Dialing Message	47
Figure 5.4.2. Update Lead Priority Dialog	47
Figure 5.4.3.. Update Lead Contacts Dialog	49
Figure 5.4.4. Update Lead Disposition Dialog	50
Figure 5.4.5. Delete Lead(s) Dialog	51
Figure 5.4.6. Print Lead(s) Dialog	52
Figure 5.5. Selling Menu	54
Figure 5.5.1. Select Prospect Dialog	55
Figure 5.5.2. Update Prospect Dialog	56
Figure 5.5.3. Update Prospect Prequalification Data Dialog	58
Figure 5.5.4. Update Prospect Citizenship Data Dialog	59
Figure 5.5.5. Update Prior Service Data Dialog	60
Figure 5.5.6. Update Law Violations Dialog	61
Figure 5.5.7. Update Personal Data Dialog	63
Figure 5.5.8. Update Test Data Dialog	64
Figure 5.5.9. Test Data/Physical Dialog	65
Figure 5.5.10. Interview Remarks Dialog	65



Figure 5.5.11. Disposition Dialog	67
Figure 5.5.12. Delete Prospect(s) Dialog	68
Figure 5.5.13. Print Prospect(s) Dialog	69
Figure 5.6. Processing Menu	71
Figure 5.6.1. Select Applicant Dialog	72
Figure 5.6.2. Update Education Dialog	73
Figure 5.6.3. Update Residence Dialog	75
Figure 5.6.4. Update Employment Dialog	76
Figure 5.6.5. Update Relations Dialog	78
Figure 5.6.6. Update Law Violations Dialog	80
Figure 5.6.7. Form 714 Request Dialog	81
Figure 5.6.8. Form 714 Transmit Dialog	82
Figure 5.6.9. Disposition Dialog	83
Figure 5.6.10. Delete Applicant(s) Dialog	84
Figure 5.6.11. Print Applicant(s) Dialog	85
Figure 5.7. Maintenance Menu	86
Figure 5.7.1. Select DEP/DTP Dialog	87
Figure 5.7.2. DEP/DTP Information Dialog	88
Figure 5.7.3. DEP/DTP Function Dialog	89
Figure 5.7.4. Update Law Violations Dialog	90
Figure 5.7.5. DEP/DTP Loss Dialog	91
Figure 5.7.6. Disposition Dialog	92
Figure 5.7.7. Delete DEP/DTP(s) Dialog	93
Figure 5.7.8. Print DEP/DTP(s) Dialog	94
Figure 6.1. RAPS Knowledge Base Context	96
Figure 6.2(a). RAPS Knowledge Base Maintenance Menu	97
Figure 6.2(b). Update Rule Dialog	98
Figure 6.2(c). Update Rule Dialog/Rule Select	99
Figure 6.5. Characters Used in Field Templates	104
Figure 6.6. Expert Inference Engine Interface Window	105
Figure 7.6. RAPS Program Architecture	112

## **1. Introduction**

The Recruiter's Applicant Processing System (RAPS) design documentation is organized into eight sections.

Section 1 is this introduction.

Section 2 introduces the initial system design concepts considered for the development of RAPS and provides a historical perspective of how these concepts guided that development.

Section 3 introduces system perceptions and requirements from different USAREC perspectives. These perspectives were gained through interviews with numerous USAREC experts and review of Department of the Army (DA) and USAREC regulations.

Section 4 identifies the general features common to the RAPS user interface. RAPS is designed as an event processing system as opposed to a sequential logic system. The events processed result from user selection of menus, entry of data into a dialog box, or selection of an option within a dialog box. The features built into RAPS capitalize on the window, menu, and dialog box capabilities supported by Arity PROLOG, and as such are standard to Arity PROLOG as well as RAPS.

Section 5 presents the system design, which is a synthesis of the many different USAREC perspectives. The resultant design is expected to be used to assist a recruiter in several phases of his/her work, such as processing leads, prospecting, selling, processing applicants, and managing individuals through DEP/DTP. For each phase, the individual menus and dialogs a user will see are completely presented along with a description of the operation of these elements. Also for each phase, the implementation considerations are provided. (It is expected that these implementation considerations will not be of substantial importance to most reviewers of the design specification. By including the implementation details for each phase, the final version of the system design specification will also serve as the long-term system maintenance documentation.)

Section 6 describes the knowledge base supporting the RAPS design. The knowledge base is derived from USAREC regulations and AR 601-210. As these regulations continue to develop and added interpretations are made, the knowledge base must be periodically updated. Facilities for updating the knowledge base are also described in this section.

Section 7 describes the RAPS software organization. RAPS was developed using Arity PROLOG and is being delivered with the full version of Arity. This section explains the files contained in both the development version and the distribution version of RAPS as well as the RAPS database structure and program architecture.

Section 8 presents the conclusions to this design documentation. Included in this section is a discussion of extensions and fine tuning for RAPS in the future; how RAPS should be deployed; and the impact RAPS can have on recruiting.

Note from STUDY PROPONENT:

*Although the initial thrust of this study was to explore automation of the central regulation governing recruiting, it became clear as the investigation proceeded that a whole process was involved. In order to actually field a program such as RAPS, we feel that two important considerations must be met. First, it is essential that such a system be provided on a separate portable machine for each recruiter. RAPS, when expanded as the contractor recommends, would essentially replace the recruiter's desk. Secondly, it is also essential that many more features, such as integrated calendar/MANDEX, would have to be incorporated. These features are key to satisfying the actual user - the recruiter - who not only must use it, but also must want to use it if the system is to be successful.*

## **2. Initial Design Concepts**

The objective of this project was to investigate the feasibility of developing an automated reasoning tool to assist Army recruiters. The tool developed, the "Recruiter's Applicant Processing System" (RAPS), is intended to assist recruiters in the application and interpretation of AR 601-210 for qualifying prospects and applicants for enlistment. RAPS employs expert systems technology and is designed to operate on MS-DOS-compatible microcomputers. RAPS is integrated with a forms-generation package that produces the enlistment contract (DD Form 1966, defined in AR 601-210). RAPS is distributable without additional licensing fees. The computer code, expert system, taxonomic system, and expert system rule base are non-proprietary and are designed to be easily accessed, updated, and maintained by USAREC personnel.

System components (the inference engine, taxonomic system, and rule base) are designed for ease of distribution through computer network systems. Data communications software with an error-checking protocol, to assure ease of distribution, is also provided.

### **2.1. Design Overview**

RAPS was designed to solve the following problems. First, the 601-210 regulations on enlistment contain a tremendous amount of knowledge. It is time-consuming to learn these regulations, and until they are learned, a recruiter's work is likely to be error-prone. Second, it is difficult for human experts to keep up with the changes in the 601-210 regulations. Third, the regulations are not organized by "events." Applicant processing, on the other hand, is organized by events. That is, applicants are processed in specific sequences, and processing sequence branching is determined by the pivotal events occurring in the sequences. (For example, a waiver for moral character must be obtained in the event a police check indicates that the applicant has been convicted of certain crimes.) Fourth, there is no convenient index to know which rules are appropriately applied in each event. Fifth, the rules are subject to some interpretation.

Recruiters understand and frequently cite these problems. Consequently, there is reason to believe that RAPS, because it focuses on these problems, will be accepted and perceived by recruiters as an important aid. The implementation and acceptance of RAPS will realize at least three benefits:

- RAPS will apply policy to concrete situations in a consistent manner.
- RAPS will make the experience of policy experts available to even the most inexperienced recruiters.
- RAPS will improve recruiting station efficiency by enabling problems and questions to be resolved rapidly by any recruiter, and by minimizing bottlenecks caused by uneven paper flow, inconsistent work habits, and "sandbagging" of applicants.

It should be emphasized that the goal of improving recruiting station efficiency can be met only if the expert system is accepted and used by recruiters. In this regard, it is especially important that RAPS is not considered another tool to enable USAREC to "micro-manage" recruiters. To avoid the "micro-management" perception, RAPS use should not be made mandatory; rather, it should be presented as an important aid to assist the recruiter with his/her work.

To counter field resistance, RAPS is designed as a full-featured system. It is designed to help recruiters process leads and prospects, and to maintain process data relevant to applicants and DEPs. In addition to codifying the 601-210 regulations and functioning as a Recruit Qualifier, RAPS includes a Form Builder for completing waivers, for requesting police records, for requesting copies of school records, etc. By allowing multiple functions that include simplifying the maintenance of LRLs, REACT files, 200 cards, etc., the system is likely to be viewed as a valuable recruiting asset instead of as another instance of recruiter "micro-management" by USAREC.

### **2.1.1. The Recruit Qualifier**

The Recruit Qualifier is designed to summarize the regulations governing eligibility criteria, policies, and procedures for enlistment and processing of applicants into the Regular Army and the U.S. Army Reserve as summarized in Army Regulation 601-210. Applicants differ, of course, in their age, sex, citizenship, physical condition, civilian occupations, moral qualifications, etc. And, according to Army Regulation 601-210, processing and documentation requirements vary with such differing applicant characteristics. The RAPS design accommodates these applicant-processing dimensions.

The Recruit Qualifier is designed with the capability to review applicant characteristics and the current state of processing and documentation to help the recruiter form a "TO DO" list of the subsequent actions necessary to qualify the recruit. The Recruit Qualifier is designed to determine waivers necessary, help automate the completion of waiver requests, guide the submission of waiver requests through proper channels, and help the recruiter predict the likelihood of success in obtaining each particular waiver.

Predicting the likelihood of success in obtaining a waiver would not require access to approval rates for waivers at company, battalion, or brigade level. Because RAPS tracks applicant actions associated with each applicant, waiver requests will also be noted. If processing of an applicant continues after a waiver was requested, then obviously the waiver was granted. Instances in which applicants are allowed to continue processing versus those instances in which applicants are prohibited from further processing can form the basis for a local database that can be used to predict the likelihood of success in obtaining a waiver.

In addition, the Recruit Qualifier is designed to improve the ability of recruiters to keep up with changes because regulation manuals are not static; rather, they tend to change on a regular basis. RAPS expert system architecture is designed to lend itself well to *effective management of change*, since the knowledge about regulations is not built into the code, as it would be if traditional programming techniques had been employed. Rather, the knowledge about regulations is contained in the knowledge base. Maintenance of the knowledge base is easier and less error-prone than maintenance (patching) of a procedural program code, or the integration of changed knowledge into human memory.

### **2.1.2. The Form Builder**

The Form Builder is designed to allow a recruiter to complete his or her DD-1966 paperwork interactively on the expert system. The Form Builder will allow maintenance of a database on the EIDS hardware to store partially completed applications. RAPS is designed with sufficient intelligence to guide the recruiter in collecting all the data necessary to complete a valid DD-1966 form while reducing the potential for errors and omissions.

In some cases, it may be easier for a recruiter to build the DD-1966 form on paper instead of on an interactive computer (e.g., when there is limited hardware available, or when paperwork is being completed out of the

office). Consequently, the Form Builder is designed to allow paper forms to be copied into the system for after-the-fact verification and error-flagging. The communications options planned for EIDS will allow completed DD-1966 forms to be transferred by modem to MEPS centers for subsequent applicant processing.

The RAPS Form Builder has been designed with the following features:

- The applicant's current address will be used as the default for address of record, parents' address, etc. The state, county, and zip code are used as defaults for all other address data (e.g., employment, school, parent/spouse address, etc.) If these are different, the recruiter can change them; if they are correct, the recruiter is able to adopt the defaults by simply pressing the Tab key.
- The Form Builder data entry screens present a logical sequence of queries expected during an applicant interview. When addressing applicant employment, the screens include all items regarding employment. When addressing education, the screens include all items relevant to schooling. This modification is considered to present an improvement over the current JOIN system, which requires that recruiters ask twice about education and employment.
- The screen driver is designed to take into account dates and recruiter entries. For example, multiple schools may be entered to summarize applicant's education, multiple residences may be entered, and also multiple employers. If, however, a recruiter enters a single high school beginning 4 years ago and if the applicant is currently a senior, it is obvious that the recruiter need not enter blanks for additional high schools. Likewise, if the first residence entry indicates that the applicant has lived there for over 5 years or since his or her 13th birthday, the screen driver does not require the recruiter to enter blanks for additional residences. Similar logic applies to employment data. In each of these cases, the recruiter is never required to enter data that is known to not be applicable. Additional entries for schools, employers, and residences that cannot be accommodated in the space allotted on the DD-1966 automatically roll over to the REMARKS fields.
- The screen driver is designed to allow simple editing within a field. Field editing functions include (1) an insert character function, (2) a delete character function, and (3) an overstrike capability.

- Print drivers are designed for the DD-398 and other relevant DD and DA forms.
- The RAPS Form Builder requires each recruiter to login. Once the recruiter is identified, all recruiter-specific information is added automatically to the Applicant Data Record (ADR) as well as to all relevant forms. This information also provides the foundation for an audit trail of actions associated with an applicant.
- All fields on each data entry screen are designed to be exited in a common fashion with the Tab key.
- Entries into the REMARKS field are made with a full field-oriented editor similar to a word processing editor that allows word-wrap.



### **3. Insights / Concerns of Individuals and Organizations**

Initial interviews with USAREC experts and review of DA and USAREC regulations identified many different perspectives on the possible nature and scope of the RAPS system. This input suggested the need to adopt a broad perspective in defining the system. This broader scope allows RAPS to address the differing sets of responsibilities and concerns expressed by the many USAREC experts, each of whom would like to see RAPS operate in a slightly different manner.

The varying system requirements regarding RAPS are summarized from the perspectives of (1) the applicant, (2) the recruiter, (3) the recruiting station, (4) MEPS, (5) USAREC, and (6) the Department of the Army.

#### **3.1. Applicant**

For an applicant, three aspects of RAPS are important. (1) RAPS forms an integral part of the initial impression regarding the Army. (2) RAPS has the potential to guide the application process in an efficient manner. And (3) RAPS helps guarantee that all of the skills and abilities of each applicant are recognized and accounted for so the applicant will "be all he or she can be."

Emphasizing today's high-tech Army. The first official contact an applicant has with the United States Army is through the recruiter and the recruiting station. It is important that this contact not negate any of the positive images associated with the Army through expensive advertising campaigns and the extensive community relations efforts advanced by recruiters.

RAPS is designed to have a sophisticated user interface that allows novice computer users to interact with the system in a professional manner. Because many young applicants are reasonably sophisticated computer users, RAPS incorporates state-of-the-art interfaces and techniques that are consistent with the image of today's high-tech Army.

RAPS automates and contributes towards a "paperless" office environment by encouraging electronic storage of applicant records. This automation allows more efficient retrieval and update of applicant information and contributes to more efficient recruiter/applicant interactions.

Streamlining and systemetizing the application process. At any moment, RAPS can retrieve an applicant's record and can determine the state of the application in the process. This ability is harnessed to streamline and

systemetize the application process. At any stage in the process, RAPS is capable of guiding the application in the direction that is most effective for closing the sale in the least time-consuming manner. This guidance is performed by maintaining "TO DO" lists for each applicant.

Because RAPS maintains applicant data as it is received, applicant interviews can proceed more efficiently. Rather than asking an applicant repeatedly for the same pieces of information, RAPS will skip data that is already known and will query only information not previously entered. This efficiency will be appreciated by both applicant and recruiter.

Helping the applicant "be all he can be." Each applicant comes to his or her recruiter with a unique combination of skills, abilities, and personal characteristics. It has remained the responsibility of the recruiter and guidance counselor to identify which of these skills, abilities, and personal characteristics might be useful to the Army. RAPS can help guarantee that applicant skills and abilities are not overlooked in the process. RAPS can help recruiters identify when special testing is called for, and RAPS can help identify civilian skills that have special relevance to the Army.

### **3.2. Recruiter**

The recruiter has many responsibilities. He or she must prospect, follow leads, recruit applicants, and maintain all of the paperwork involved in the process. Simultaneously, the recruiter must satisfy recruiting goals as specified by Mission Box. All of these activities must be carried out according to regulations and with integrity.

The most common complaint we have heard in our previous work with USAREC recruiters is that there is not enough time to complete all of the requirements in a 40-hour work week. Consequently, RAPS focuses on labor- and time-saving features that will have a significant impact on recruiter time management.

Paperless processing. RAPS is designed to have a major impact on the paper-handling requirements of recruiting. RAPS can maintain lead lists, LRLs, 200 cards, and can generate the DD-1966 and related forms, including requests for school and police records, waiver requests, and MEPS forms. RAPS is designed with sufficient intelligence that it never needs to have data entered more than once. As leads become prospects, their data remain available to the system. As prospects become applicants, the same data remain accessible. And as applicants become recruits and DEP in, their data

continue to be maintained on RAPS. Consequently, whenever it is necessary to generate a form or list of applicants at a given stage of processing, the RAPS data may be used for this purpose. The recruiter never needs to retype data.

Mission Box. Mission Box is the most emphasized aspect of recruiting. Recruiters constantly face Mission Box requirements in prioritizing their activities. Because RAPS maintains complete applicant data, it can be used to coordinate and schedule applicant processing to meet Mission Box requirements. RAPS can instantly access the number of Category A males, females, high school grads, seniors, etc. to help recruiters prioritize lead-generation and prospecting activities. Additionally, RAPS can help schedule the orderly processing of leads, prospects, and applicants to take into account Mission Box requirements.

Scheduling to meet Mission Box should become more flexible because RAPS is designed to help maintain a steady stream of leads, prospects, and applicants in the recruiting pipeline. Similarly, an even flow in the pipeline should eliminate the necessity as well as the pressure to sandbag applicants for future needs. Instead, scheduling will reflect current mission requirements and applicant availability, and RAPS will direct and guide current lead generation and prospecting to meet future mission requirements.

Another significant aspect of recruiting regards the documentation of activities associated with Mission Box. RAPS allows command to easily review records of telephone contacts, appointment records, and records of interview results. The time and effort associated with maintaining the MANDEX is addressed directly by RAPS, which automatically logs all contacts and the results of these contacts. These records can be used to develop the reports and documentation required by command. In addition, these records form the foundation for an audit trail of activity associated with each applicant.

Regulations. Understanding and complying with Army regulations is difficult enough. It is doubly difficult because Army regulations are in a constant change of flux, and because USAREC policy as applied to Army regulations also changes. RAPS codifies the knowledge in AR 601-210 and allows its systematic application by recruiters. The consequences of this codification include the following: (1) even novice recruiters will process applicants in accordance with regulations; (2) changes in AR 601-210 will be instantly and uniformly applied throughout USAREC; and (3) USAREC policies, interpretations, and clarifications will be consistently applied

without the need to repeatedly check with headquarters Plans and Policy Division on the same matters.

Integrity. Lapses in recruiting integrity are typically attributed to pressure. RAPS is designed to help alleviate pressure by freeing recruiters from day-to-day paperwork requirements and allowing them to spend more time on productive (and enjoyable) lead generation and prospecting. By structuring recruiter efforts and helping recruiters manage their time more efficiently, job pressure is reduced and lapses in integrity should abate.

### **3.3. Recruiting Station**

The issues affecting the recruiting station are very similar to those affecting the recruiter. In addition, recruiters are managed at the station level, where the Production Management System (PMS) is maintained and implemented. RAPS has the potential to significantly assist in station management. RAPS will also assist the station commander in managing applicant throughput to make station mission.

PMS. PMS is an individualized, goal-setting and goal-management system that leads to Mission Box attainment. At the heart of this system is the recognition that individual differences exist among recruiters. Some recruiters will require fewer leads to generate a sale, while others may require significantly more leads. PMS sets individual goals for each recruiter and specifies the number of appointments to be made, the number of appointments to be conducted, the number of applicants to be tested, etc. These goals are set according to the historical pattern established by each recruiter and help assure Mission Box accomplishment.

Unfortunately, the data-collection effort associated with PMS is not trivial. To be useful, the historical data for each recruiter must be accurate. And this accuracy can only be obtained through the detailed chronicling of each recruiter's daily activity. RAPS significantly reduces this record-keeping requirement because it automatically and painlessly logs each applicant contact into the applicant database. PMS data are electronically available for future projections.

The historical data available through RAPS also allows recruiters to compare their own performance histories with the performance of other recruiters. Given this opportunity, recruiters are likely to compare and reassess their work habits to improve conversion rates. Working smarter becomes easier

with reference to accurate data about one's own performance and how it compares to that of others.

**Making station mission.** The same data that are relevant to accurately project recruiter accomplishments are also useful for projecting station effort to make mission. The station commander can have, at any moment, a complete summary of where the station stands with respect to appointments made, conducted, applicants tested, etc. This information is critical for managing recruiters with the goal of making station mission. RAPS data can help assign immediate performance objectives with the purpose of managing the workload so that mission is accomplished by the end of the mission month.

### **3.4. MEPS and METS**

The interaction between recruiters and the Military Enlistment Processing Stations (MEPS) and Military Enlistment Testing Stations (METS) is guided by recruiter-generated paperwork. Form 714 must be generated for testing, and enlistment packets must be generated for enlistment. MEPS and METS must be informed regarding the number and nature of candidates appearing on the floor each day.

RAPS is designed to facilitate the interaction between recruiters and MEPS and METS in two ways. First, RAPS automates the projection process, assuring that accurate and timely projection data are available for MEPS and METS. Second, RAPS does initial quality control on enlistment packets to assure that all of the information critical to successful processing is included.

**Projections.** RAPS maintains the complete status of each prospect. When all of the information necessary for a prospect is available and the prospect can be sent for testing, RAPS notifies the station commander in a report. In addition, RAPS makes projections to the station commander regarding when applicants are likely to be ready for MEPS processing. These reports allow the generation of accurate projections to MEPS and METS.

The projections themselves are made electronically through ARADS. RAPS incorporates communications software that passes relevant information from the Applicant Data Record (ADR) to ARADS through a modem. RAPS verifies the transfer of projection information.

**Quality control.** The knowledge base, which includes Army Regulation 601-210 and current USAREC policies, guides RAPS and the recruiter in processing applicants. RAPS assures that (1) all relevant data and (2) all

relevant forms are completed in each enlistment packet. In addition, RAPS assures that processing is done according to current Army regulations and current USAREC policy. This quality control assures that applicants do not need to be sent back to recruiting stations for additional processing and that processing proceeds in a predictable manner.

In addition, complete and accurate enlistment packets assure that MEPS guidance counselors have all the information necessary to enlist applicants into MOSs that can capitalize on the skills and abilities of the applicants. By assuring that all relevant applicant information is considered at the MEPS, counseling can simultaneously serve both the applicant and the Army.

### **3.5. USAREC**

USAREC benefits from RAPS by being able to develop an accurate database that relates all PMS activities to actual mission accomplishment. The distribution of PMS activities allows more accurate projection of mission and the ability to affect mission accomplishment by manipulating PMS.

RAPS also provides USAREC with a means to develop the distribution of applicant characteristics at every stage in the recruiting process (initial interview, sales interview, testing, etc.). Accurate distributions of applicant characteristics at these different points allow modeling and simulation to determine the effects of USAREC regulations and policies on mission accomplishment.

Currently, the Plans and Policy Division of Recruiting Operations frequently field the same questions on policy from many different recruiting stations and many different recruiters. Even when the Plans and Policy Division sends memos clarifying Army or USAREC regulations, recruiters who do not see these directives, or who forget their contents, may repeatedly call headquarters for clarification. RAPS allows USAREC directives and clarifications to be codified in the knowledge base, where they can be accessed when needed and appropriately implemented as required.

Finally, RAPS provides a database of recruiter activities as they relate to all stages of recruiting. This database allows inspection and oversight of recruiter operations.

### **3.6. Department of the Army**

Department of the Army has as its major concern the consistent implementation of AR 601-210 for recruiting. The RAPS knowledge base offers this consistent implementation as a central feature.

### **3.7. Summary**

The very different perspectives and insights offered by USAREC staff suggested the need to develop a broad, fully featured processing system for recruiters. This broad perspective, used in building RAPS, assures that RAPS significantly addresses the needs of recruiters. Moreover, recruiters who find that the features of RAPS are helpful and interesting are more likely to use, experiment with, and expand their use of RAPS.

## **4. General RAPS Features**

RAPS is designed as an event processing system as opposed to a sequential logic system as with traditional computer applications. The events processed result from recruiter selection of menus, entry of data into a dialog box, or selection of an option within a dialog box. The features built into RAPS capitalize on the window, menu, and dialog box capabilities supported by Arity PROLOG, and as such are standard to Arity PROLOG as well as to RAPS. The following paragraphs provide an introduction to the operation of the elements that provide the foundation for the design descriptions presented in Section 5.

### **4.1. Windows**

A window is a designated screen area in which all current screen operations are performed. A window may be the size of the whole screen or a portion of the screen. Only one window can be the current window. When a window is the current window, it appears in the foreground, and all screen management and input/output operations refer to this window.

RAPS employs different windows for different functions. Windows may be characterized by color, position, size, border, and content. These features are managed by RAPS to make it easy and intuitive to move from one function to another within RAPS, as well as to execute the functions possible from each window.

### **4.2. Menus**

A menu is a label at the top of a window that is used to present one or more options for user selection. Pressing the Esc key will transfer a user to menu mode. When in menu mode, the right-arrow and left-arrow keys may be used to move from menu to menu.

In addition to choosing a menu through the use of the Esc key, it is possible to choose a menu through the use of a hot-key. A hot-key is a key defined to be associated with a particular menu. The associated hot-key is one of the characters in the menu label (often the first character). When a menu is associated with a hot-key, then the character representing the hot-



key appears in a different color than all the other characters in the menu label.

To use a hot-key, hold down the Alt key while pressing the desired hot-key. This brings the control associated with the hot-key into focus.

With a specific menu selected, the down-arrow key will cause the items below the menu to be displayed. With these items displayed, the user may use the down-arrow and up-arrow keys to position to the desired option. The current menu option is highlighted. The highlighted option is selected using the enter key.

There may be hot-keys associated with various options on a menu. If present, these will be displayed in a different color than the rest of the menu option text. These hot-keys differ from the menu selection hot-keys. A user may simply press the appropriate key to select the option. It is not necessary to press the Alt key in conjunction with the hot-key as with menu selection hot-keys.

### **4.3. Dialog boxes**

A dialog box is a special type of window that can contain a number of "controls" for performing various functions. As the name implies, a dialog box is intended to provide a "dialog" between the system and the user whereby the user is prompted to make certain choices or provide certain information.

A dialog box can contain any number of different types of controls. Each control is used to gather information in a different way. For example, an edit field is a control in which a user types something, whereas a choice button control lists an option that the user can choose by pressing the space bar, which makes an "X" appear next to the chosen item. A control is said to be in "focus" when the cursor resides in that control. Anything typed at the keyboard while a control is in focus applies to that control. Arity/PROLOG provides eight standard controls. RAPS uses them all in combinations that suit each function.

There are two types of dialog boxes: stay-up and pop-up dialog boxes. Whenever a dialog box is created, a window is also created, and the dialog box is displayed in that window. A stay-up dialog box is displayed in a stay-up window; a pop-up dialog box in a pop-up window. When a stay-

up dialog box is exited, the dialog box continues to exist on screen. The window that was current before the dialog box was displayed is made the new current window. When a pop-up dialog box is exited, the dialog box and associated window are deleted and control returns to the window that was current before the dialog box was displayed.

#### **4.4. Dialog box Elements**

Dialog boxes can vary with the same attributes as windows. In addition, dialog boxes can contain a variety of different controls that can be positioned anywhere within the dialog box. The standard controls that are used by RAPS include:

##### **4.4.1. Choice Buttons**

A choice button lists a single item. A user can choose this item by moving the cursor to the choice button and pressing the space bar. This causes an "X" to appear in the choice bracket, indicating that the item has been selected. If the space bar is pressed while the cursor is at a choice button that has already been selected, then the "X" will be removed (deselected), indicating that the item is no longer chosen. Choice buttons can be used singularly or in groups.

##### **4.4.2. Radio Buttons**

Radio buttons are used when there are two or more options from which the user can choose only one option. Radio buttons derive their name from car radio-tuning buttons that are used to tune into a single station at a time. A radio button is chosen by bringing the button into focus.

Choice buttons or radio buttons are sometimes "greyed." When a greyed item appears in a dialog box, it cannot be chosen by the user. This is useful in instances where the same dialog box is used a number of times, but where different subsets of choices are valid from one time to another.

#### **4.4.3. Edit Field**

An edit field allows the user to enter a single line of text into a field. An edit field is horizontally scrollable and accepts up to 200 characters.

#### **4.4.4. Push Button**

A push button performs a singular action. Push buttons are often used for confirming an action, such as having to push an "OK" button to indicate that some desired change should take place. When using push buttons, one button is frequently defined as the default button. If an Enter key is pressed anywhere in the dialog box (other than within an edit box or region), then the action associated with the default push button is performed.

#### **4.4.5. List Box**

A list box presents a vertically scrollable list of items from which the user can choose. There are two types of list boxes: radio and choice list boxes. Only one item can be chosen from a radio list box, whereas the user has the option to choose any number of items from a choice list box.

#### **4.4.6. Edit Box**

An edit box is a box that contains a fully functional editor. The edit box includes a border and a cursor that indicates the row and column the cursor is at. There are two types of edit boxes: read/write and read-only. In a read/write edit box, it is possible to both read and alter the contents of the edit box. In a read-only edit box, it is possible to only read the contents.

#### **4.4.7. Text**

Text controls are frequently used within dialog boxes to display information. Such text is not used for offering choices.

## **4.5. Moving Through and Selecting from Dialog Boxes**

Each control is considered a separate entity in a dialog box. Certain controls, such as choice buttons, can be used singularly or in groups. To move forward from one control or group of controls to another, press the Tab key. To move back from one control or group of controls to another, press Shift+Tab.

Once the cursor has been moved to a particular control, the movement of the cursor within that control depends on the type of control. The following is a list of controls and keys used for cursor movement and item selection within controls when a control is in focus.

### **4.5.1. Choice Button**

The space bar acts as a toggle for selecting and deselecting an item. An "X" appears next to the item when it is chosen. When choice buttons are grouped, one can move among choice buttons by using the Up-arrow, Down-arrow, Left-arrow, and Right-arrow keys.

### **4.5.2. Radio Button**

A radio button is chosen when it is brought into focus. Since radio buttons always occur in groups of two or more, the group cursor movement rules apply. Thus, it is possible to move among radio buttons using the Up-arrow, Down-arrow, Left-arrow, and Right-arrow keys.

### **4.5.3. Edit Field**

If an edit field is empty, then pressing a character will insert that character in the edit field. If there is text in an edit field, then the text is automatically "selected" and the cursor is placed at the end of the selected text. If a character is typed, the selected text is erased and the typed character is displayed as the first character in the edit field. Pressing the Backspace key as the first key after moving into an edit box containing text will move the cursor to the left, erase one character, and deselect the

text in the edit field. The following keys perform the indicated actions in the edit field:

Right-arrow/Left-arrow - Pressing the Right-arrow key moves the cursor one character to the right. Pressing the Left-arrow key moves the cursor one character to the left.

Home/End - The Home key moves the cursor to the first character in the edit field. The End key moves the cursor to the last character in the edit field.

Del - If no text is selected, deletes the character at the current cursor position. If text is selected, deletes the selected text.

Backspace - Deletes the character to the left of the current cursor position.

Shift+Del - Deletes the selected text.

Shift+(direction keys) - Select text by holding down the Shift key and pressing the direction keys: Left-arrow, Right-arrow, Home, End, Ctrl-Left-arrow, Ctrl-Right-arrow. Selection is made from the current cursor position.

#### **4.5.4. Push Button**

While in focus, pressing the Enter key or space bar will select the button in focus.

#### **4.5.5. List Boxes**

There are two types of list boxes: radio and choice list boxes. The Up-arrow and Down-arrow keys move the cursor between choices in list boxes. The PgUp key scrolls the box items up one page. (A page is defined as the number of items that are displayed in the box at one time.) The PgDn key scrolls the items down one page. The Home key moves the cursor to the first item, and the End key moves the cursor to the last item. For radio list boxes, the selected item is the item where the cursor is positioned. For choice list boxes, pressing the space bar acts as a toggle for selecting and deselecting each item.

#### 4.5.6. Edit Box

There are two modes of operation for an edit box while it is in focus: movement and edit mode. When an edit box is first brought into focus, it is in "movement" mode. An edit box is in movement mode when the cursor shape changes to a block shape. While in movement mode, pressing the Tab key moves to the next control, and pressing Shift+Tab moves to the previous control. Pressing any other key switches to "edit" mode. Edit mode allows the user to use the box as an editor. The movement and manipulation keys while in edit mode are:

Right-arrow/Left-arrow - Pressing the Right-arrow key moves the cursor one character to the right. Pressing the Left-arrow key moves the cursor one character to the left.

Up-arrow/Down-arrow - Pressing the Up-arrow key moves the cursor up one line. Pressing the Down-arrow key moves the cursor down one line.

Home/End - The Home key moves the cursor to the first character in the current line. The End key moves the cursor to the last character in the current line.

PgUp - Up one page.

PgDn - Down one page.

Ctrl+Left-arrow - Left one word.

Ctrl+Right-arrow - Right one word.

Ctrl+Home - Beginning of text.

Ctrl+End - End of text.

Ctrl+g - Goes to the line specified. If no text is selected, deletes the character at the current position. If text is selected, deletes the selected text.

Backspace - Deletes the character to the left of the current cursor position.

Shift+(direction keys) - Select text by holding down the Shift key and pressing the direction keys: Up-arrow, Down-arrow, Left-arrow, Right-arrow, Home, End, Ctrl-Left-arrow, Ctrl-Right-arrow. Selection is made from the current cursor position.

Shift+Del - Deletes the selected text.

Shift+Tab - Moves to the previous control.

Tab - Moves to the next control.

#### **4.5.7. Hot-Keys**

In addition to choosing a dialog box item through the use of the Tab and Shift+Tab keys and the associated movement keys for a control, it is possible to choose an item through the use of a hot-key. A hot-key is a key that is defined to be associated with a particular dialog box control. The associated hot-key is one of the characters in the item label (often the first character). When a dialog box control is associated with a hot-key, then the character representing the hot-key appears in a different color than all the other characters in the item label.

To use a hot-key, hold down the Alt key while pressing the desired hot-key. This brings the control associated with the hot-key into focus.

#### **4.5.8. Exiting a Dialog box**

If a dialog box does not contain any push buttons, it can be exited at any time by pressing the Enter or Esc keys (unless one is typing in an edit box or edit region, in which case you must first press the Tab key).

If the dialog box contains push buttons, then one of the push buttons may be defined as the default button. When a push button is brought into focus, it automatically becomes the default button. Pressing the Enter key at any time while in the dialog box causes the action specified by the default button (if one exists).

## **5. RAPS Design / Operation**

The following RAPS design is presented in the sequence in which the system is expected to be used to assist a recruiter in processing leads, prospecting, selling, processing applicants, and finally, managing individuals through DEP/DTP.

Figure 5 summarizes the RAPS design. The figure illustrates how RAPS addresses all of the significant functions of a recruiter's job: lead generation, prospecting, selling, processing, and DEP management. Figure 5 suggests that each of these functions can be performed in a "stand-alone" fashion. That is, RAPS will allow a recruiter to perform lead-generation activities only. Or, a recruiter can use RAPS solely for DEP/DTP maintenance, prospecting, selling, or processing. In addition, each of these major functions can proceed sequentially according to the events that define the different functions. In this instance, RAPS maintains leads. Then, when a lead agrees to a sales interview, RAPS maintains prospect information and supports the sales interview. After the applicant commits to processing, RAPS assists in preparations for testing and enlistment packet development. If the applicant enlists, RAPS functions to organize and include the applicant in DEP functions.

### **5.1. RAPS Windows**

RAPS uses five different kinds of windows (Figure 5.1): the large Main Window, the Menu Bar across the top of the screen, the Status Line at the bottom of the screen, various Pull-Down Menus, and various dialog boxes.

Only one window is active at a time. The active window is the window in which the cursor resides. And within any active window area, only commands relevant to that active window are functional.

In general, the Main Window is used to communicate RAPS activities to the user. The Menu Bar is used to organize the main RAPS functions. The Pull-Down Menus are used to access the capabilities under each major function. The dialog boxes are used to display messages and to accept user input. The Status Bar is used to display information relevant to the current session.



This page is replaced by the diagram "5. Design/Operation" done in Inspiration. This page is here so the auto table of contents will pick up the Figure 5. label and put it in the List of Figures.

When a new page is printed print the file "5. Figure of Design" and use the page number off this page. After printing that page put it back in the LaserWriter and print the diagram from Inspiration. Put paper back in LaserWriter tray "Footer first - face up" (this needs to be checked).

Figure 5. RAPS Design Concept

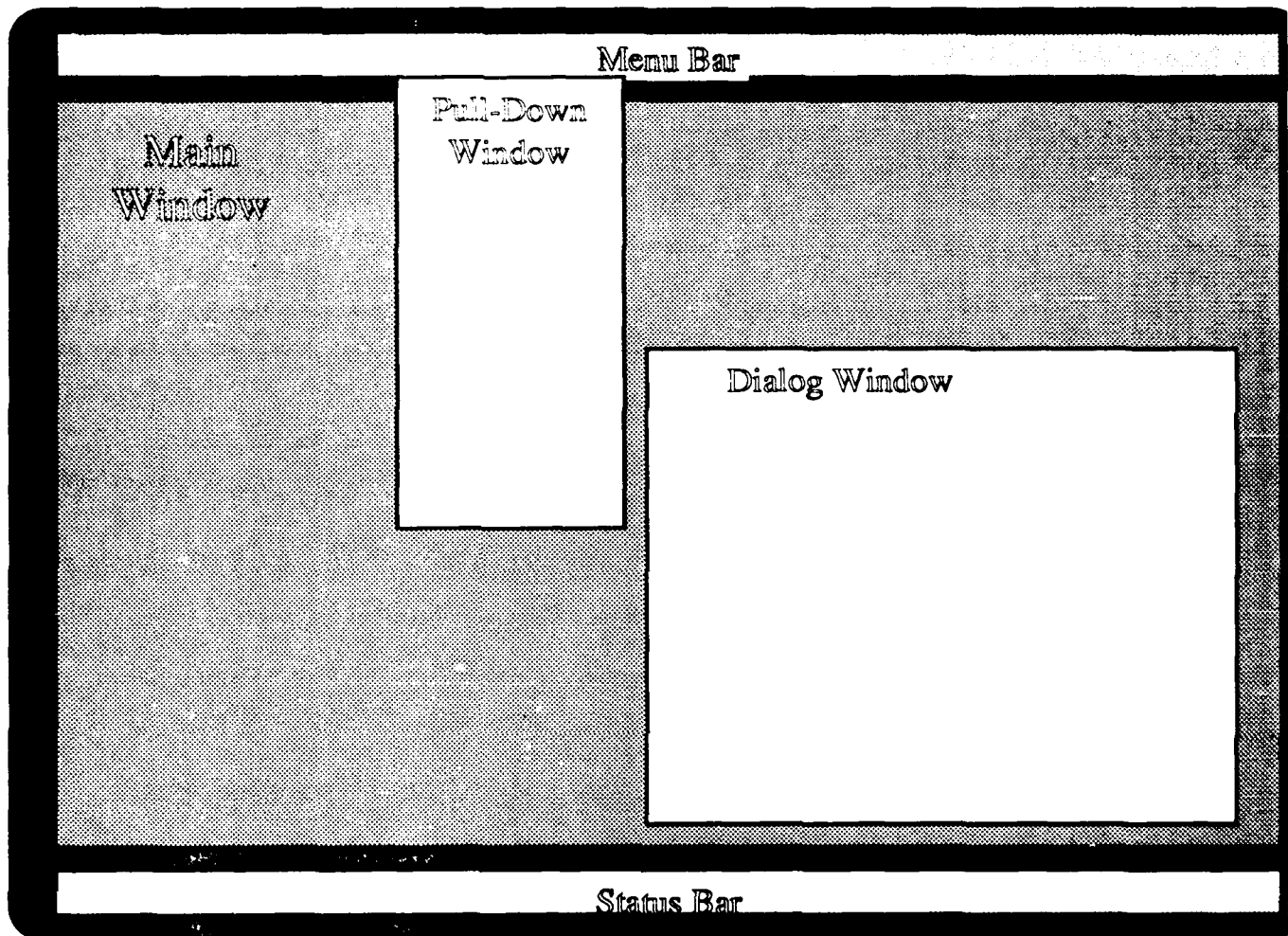


Figure 5.1. RAPS Windows

In the sections that follow, the menu bars and dialog boxes relevant to RAPS features and functions are presented.

### 5.1.1. RAPS Menu Bar and Pull-Down Menus

RAPS begins with a short message in the Main Window. And the Main Window remains active until (1) the user types the F1 function key to access help, (2) the user types Shift-F10 to quit, (3) the user presses the Alt key to activate the Menu Bar, or (4) the user presses one of the hot-keys defined on the Menu Bar.

Pushing the Alt key moves the cursor from the Main Window to the Menu Bar and activates the Menu Bar. Once on the Menu Bar, it is possible to use the Left- and Right-arrow keys to shift from one menu function to another. Shifting from one function to the other with the arrow keys highlights the active function. Pressing the Enter key while a function is activated pulls down the Pull-Down Menu associated with that function.

Recruiter	Lead	Prospecting	Selling	Processing	Maintenance	KBM	Help
-----------	------	-------------	---------	------------	-------------	-----	------

Figure 5.1.1(a). RAPS Menu Bar

Eight RAPS functions are defined on the Menu Bar: (1) Recruiter Function, (2) Lead Function, (3) Prospecting Function, (4) Selling Function, (5) Processing Function, (6) Maintenance Function, (7) Knowledge Base Maintenance Function, and (8) Help Function. These functions can be accessed by using the arrow keys to highlight a selection and then pressing the Enter key or by pressing the corresponding hot-key.

Hot-keys are defined as Alt+(key). The keys corresponding to each of the eight RAPS functions are defined in red on the Menu Bar. (Figure 5.1.1(a). above defines the hot-keys in ~~outline~~ form.) To gain immediate access to the Maintenance Functions that are presented on the associated Pull-Down Menu, for example, simply press Alt-M while in the Main Window or anywhere on the Menu Bar.

Select...
DEP Information...
DEP Function...
Law Violations...
DEP Loss Data...
Disposition
Delete...
Print...

Figure 5.1.1(b). RAPS Pull-Down Menu

To exit from any Pull-Down Menu and return to the Main Window, simply press the Esc key. To change from one Pull-Down Menu to any other Pull-Down Menu, simply use the Left- or Right-arrow keys to switch from one set of RAPS functions to another.

### 5.1.2. Menu Greying

The RAPS knowledge base is used to determine which functions are valid at a given time and for a given lead, prospect, applicant, or recruit. Invalid functions are simply turned off by greying the menu areas that are inappropriate for a given prospective recruit, thus denying access to those areas. Only those functions that are not greyed (that is, are legal) are allowed.

### 5.1.3. Menu Checking

To assist the recruiter in tracking progress with leads and applicants, RAPS checks off the menu items that have been accessed and where changes in the applicant's data record have been made. A small check mark appears at the left of any RAPS function addressed (Figure 5.1.3.).

Select...
√ DEP Information... DEP Function...
√ Law Violations... DEP Loss Data...
Disposition
Delete... Print...

Figure 5.1.3. RAPS Menu Checking

#### 5.1.4. RAPS Help

RAPS offers two Help files for users, both of which are loaded into the RAPS database and are seamlessly integrated with the RAPS system. The first file is a Menu Help file; this introduces all of the menu options available. The second is a Dialog Help file; this introduces all of the dialog options. Either of these two files is accessible from the Menu Bar by scrolling to "Help" and selecting either "Menu Help" or "Dialog Help" from the Pull-Down Menu.

Help is displayed in a Help window [Figure 5.1.4(a).] that overlays the currently active window. The window presents textual help along with a blinking cursor. The cursor may be scrolled up or down through the text using the Up- and Down-arrow keys or the PgUp and PgDn keys so that all of the current Help file is available once Help is accessed.

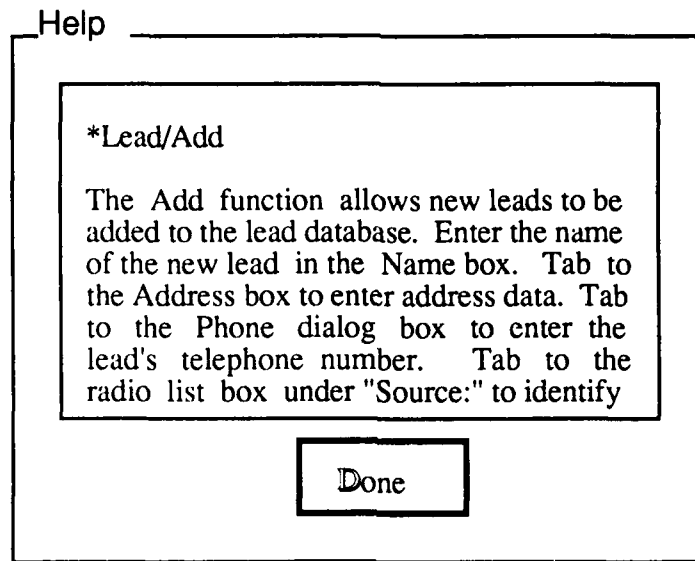
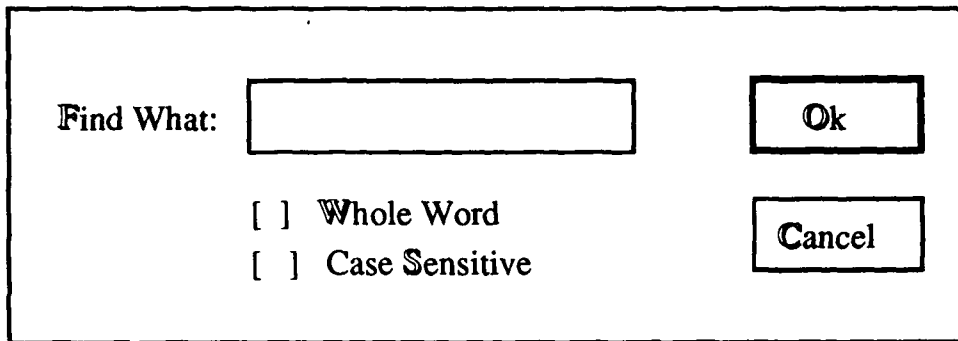


Figure 5.1.4(a). RAPS Help

To quit from the Help window and to return to the point from where Help was accessed, simply press the Esc key or TAB to the "Done" radio button at the bottom of the Help window and press the Enter key.

RAPS offers access to Help from both the Main Window and from any dialog box by pressing the function key F1. Pushing F1 while in a dialog box presents context-sensitive help specific to the current dialog box.

Once Help has been accessed, RAPS offers a search capability through the Help file to access a specific topic. While Help is displayed, press F4; another pop-up dialog appears into which you can enter any text string in the box labeled "Find What" [Figure 5.1.4(b).] RAPS then searches the Help file for that string, and if found, displays that section of the Help file in the Help window. If the string is not found, RAPS displays the message "STRING NOT FOUND."

A dialog box titled "Find What:" with a text input field. Below the input field are two checkboxes: "[ ] Whole Word" and "[ ] Case Sensitive". To the right of the input field are two buttons: "Ok" and "Cancel".

Find What:

☐ Whole Word

☐ Case Sensitive

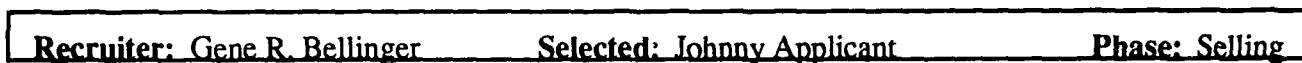
Ok

Cancel

Figure 5.1.4(b). RAPS Find Help

### 5.1.5. RAPS Status Bar

The Status Bar at the bottom of the screen is used to remind the user of the current session. Information is presented regarding the recruiter, the selected record, and the phase of processing (Figure 5.1.5.).

A horizontal status bar with three sections: "Recruiter: Gene R. Bellinger", "Selected: Johnny Applicant", and "Phase: Selling".

Recruiter: Gene R. Bellinger      Selected: Johnny Applicant      Phase: Selling

Figure 5.1.5. RAPS Status Bar

The Status Bar allows the user to determine at a glance which menu options are valid and which records are accessible.

### 5.1.6. RAPS Date;Time Stamp

Some of the RAPS dialogs require the entry of date and time. These entries are especially important for maintaining data about the frequency of lead contacts. In addition, such data have relevance for entries in the MANDEX. The F8 function key may be employed to automatically enter the RAPS system date and time for these purposes.

Whenever the user is entering data in an edit field, the F8 key will enter the system date and time in the format mm/dd/yy;hh:mm into the current field.

## 5.2. Recruiter Access Phase

Because RAPS may be used by many recruiters, the system must link individual applicant data with the recruiter responsible for that applicant. The login process identifies the recruiter to the system. All information processed by the recruiter is then tagged to indicate which recruiter is responsible for the information.

The recruiter login process is chosen from the main Menu Bar by highlighting the "Recruiter" option and then pressing the Enter key. The "Recruiter" option may be chosen by moving the highlight with the arrow keys on the keyboard or by typing the hot-key, "R." while the main Menu Bar is active.

### 5.2.1. Recruiter Menu

When the "Recruiter" option is chosen from the main Menu Bar, a Pull-Down Menu appears that presents six options: four options for identifying recruiters; one option for saving the database; and another, quit, for exiting from RAPS (Figure 5.2.1).

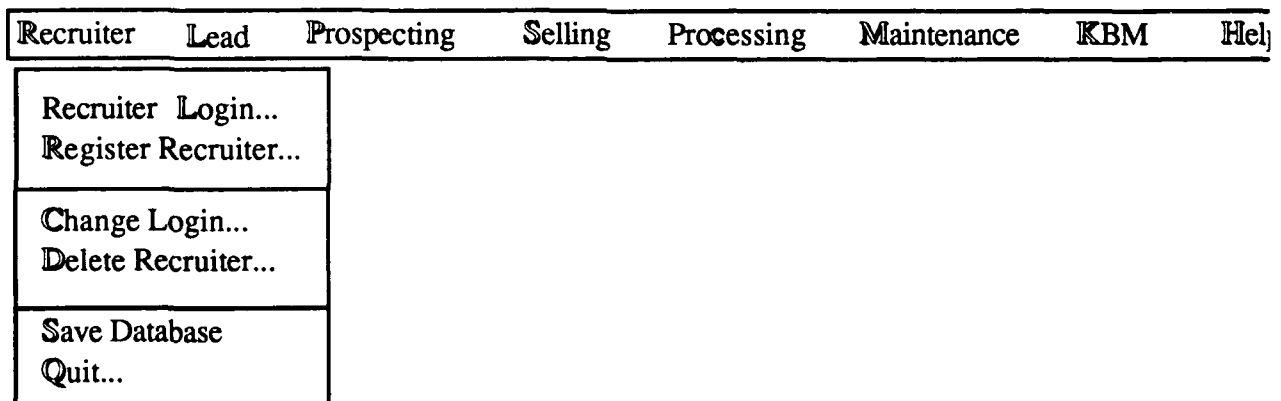


Figure 5.2.1. Recruiter Pull-Down Menu

Use the options in the Recruiter Pull-Down Menu to identify yourself to the RAPS system. Identification is necessary to assure that your work is credited to your applicants. "Login" allows you to identify yourself to RAPS if you have previously been registered as an authorized RAPS user. "Register" allows you to identify yourself as an authorized RAPS user. You may also use the Register dialogs to edit data associated with the "logged recruiter."



"Change" allows you to process an applicant for another recruiter and thereby credit him/her with your work. "Delete" allows you to delete a recruiter from the RAPS system and thereby deny that recruiter further access to the applicant database.

The first option, "Recruiter Login," is used to identify an already registered recruiter to the system. This option may be chosen by moving the highlight to "Recruiter Login" with the Up- and Down-arrow keys and pressing the Enter key. Alternatively, "Recruiter Login" may be chosen by pressing the hot-key, "L."

The second option, "Register Recruiter," is used to introduce new recruiters to the system. This option is usually chosen only once for each recruiter. It may be selected by moving the highlight to "Register Recruiter" with the Up- and Down-arrow keys and pressing either the Enter key or the corresponding hot-key, "R."

The third option, "Change Login," is used to allow access to the records of another recruiter. Choose this option by moving the highlight to "Change Login" with the Up- and Down-arrow keys and pressing either the Enter key or by pressing "C," the corresponding hot-key.

The fourth option, "Delete Recruiter," is used to allow the deletion of a recruiter record. Choose this option by highlighting the choice and pressing either the Enter key or the hot-key "D."

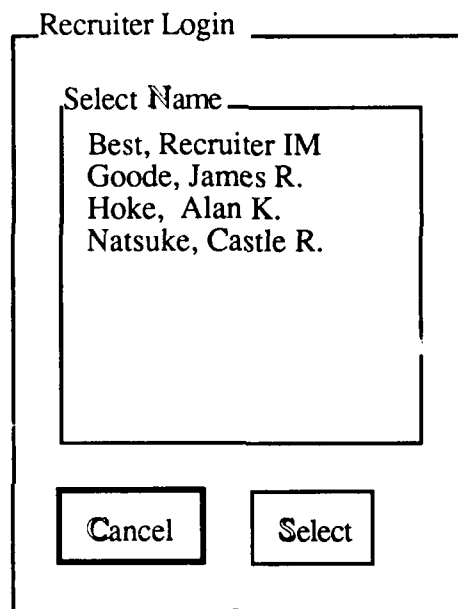
Two additional options are presented on the Recruiter Pull-Down Menu: "Save Database" and the "Quit." These additional options are explained below.

RAPS does not update its databases with every action. Instead, changes are stored temporarily and are implemented on the permanent database periodically as well as upon exit from the RAPS system. Under some circumstances, it may make sense to assure that changes have been recorded to the permanent database. Choose the option "Save Database" for this purpose. Again, this option may be chosen by highlighting the choice and pressing the Enter key or the hot-key "S."

The last option, "Quit," is used to exit from RAPS. Choose this option by highlighting the choice and pressing either the Enter key or the hot-key "Q". Remember that it is also possible to quit from any point in RAPS by pressing the hot-key Shift-F10.

### 5.2.2. Recruiter Login Dialog

The Recruiter/Login dialog (Figure 5.2.2.) presents a radio list box in which only one name out of many may be selected. Find your name in the box and select it by moving to it with the Up or Down arrow key. The name selected is marked by an arrowhead.



The diagram shows a rectangular dialog box titled "Recruiter Login". Inside the box, there is a label "Select Name" followed by a list box containing four names: "Best, Recruiter IM", "Goode, James R.", "Hoke, Alan K.", and "Natsuke, Castle R.". Below the list box, there are two buttons: "Cancel" on the left and "Select" on the right.

Figure 5.2.2. Recruiter Login Dialog

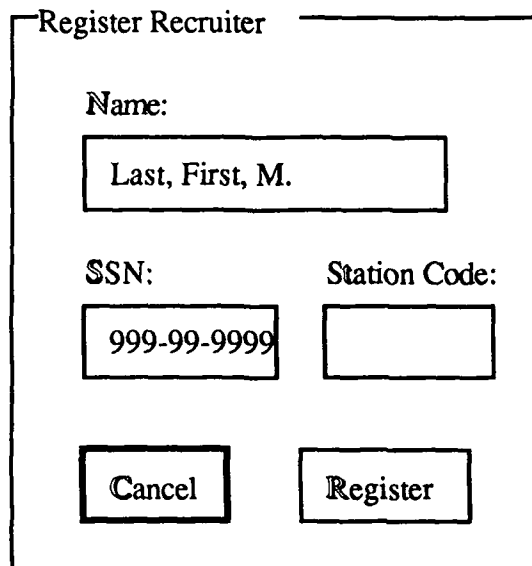
Having selected your name, use the Tab key to move to the "Select" radio button. Pressing the Enter key at this point will identify the selected recruiter to RAPS. The hot-key Alt-S may also be used to identify the selected recruiter to RAPS.

Pressing the hot-key Alt-C will cancel the selection. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the selection.

### 5.2.3. Register Recruiter Dialog

The Recruiter Register dialog (Figure 5.2.3.) presents the name, Social Security number, and station code fields for identifying new recruiters to

RAPS. Fill in the recruiter's name, then press the Tab key to go to the SSN field and enter the SSN. Next TAB to the Station Code field and fill in the station code. Finally, TAB to highlight the Register radio button and press the Enter key. This will register the new recruiter. Pressing the hot-key Alt-R will also register the new recruiter.



The dialog box is titled "Register Recruiter". It contains three input fields and two buttons. The first field is labeled "Name:" and contains the text "Last, First, M.". The second field is labeled "SSN:" and contains the text "999-99-9999". The third field is labeled "Station Code:" and is empty. At the bottom, there are two buttons: "Cancel" and "Register".

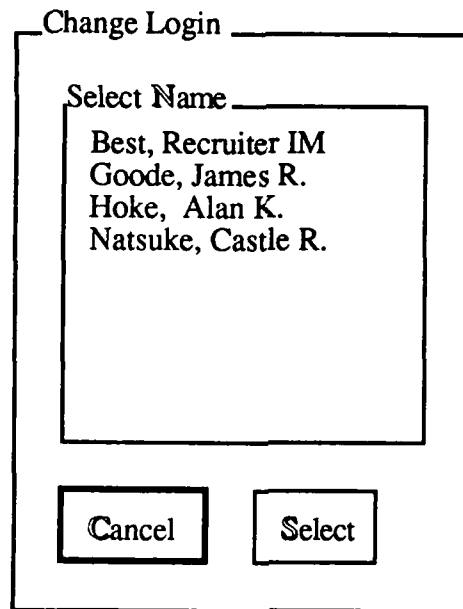
Register Recruiter	
Name: Last, First, M.	
SSN: 999-99-9999	Station Code: 
Cancel	Register

Figure 5.2.3. Register Recruiter Dialog

Cancelling an erroneous registration is done by pressing the hot-key Alt-C or press the Enter key while the "Cancel" radio button is selected.

#### 5.2.4. Change Login Dialog

To allow one recruiter to process data for another or to allow a station commander to review individual recruiter data, the Change Login option allows the selection of another recruiter as the current recruiter. This dialog operates the same as the Recruiter Login dialog.

A rectangular dialog box titled "Change Login". Inside, there is a label "Select Name" above a list box. The list box contains four names: "Best, Recruiter IM", "Goode, James R.", "Hoke, Alan K.", and "Natsuke, Castle R.". At the bottom of the dialog box are two buttons: "Cancel" on the left and "Select" on the right.

Change Login

Select Name

Best, Recruiter IM  
Goode, James R.  
Hoke, Alan K.  
Natsuke, Castle R.

Cancel Select

Figure 5.2.4. Change Login Dialog

The Recruiter/Change dialog (Figure 5.2.4.) presents a radio list box in which only one name out of many may be selected. Find the name of the recruiter whose work you wish to complete by moving to it with the Up- or Down-arrow key.

Having selected a recruiter's name (marked by an arrowhead), use the Tab key to select the "Select" radio button. Pressing the Enter key at this point will identify the selected recruiter to RAPS. The hot-key Alt-S may also be used to identify the selected recruiter to RAPS.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the selection.

#### 5.2.5. Delete Recruiter Dialog

The Delete Recruiter dialog (Figure 5.2.5.) presents a choice list box in which one or more recruiter's(s') name(s) may be selected. Find the name(s) of the recruiter(s) whose work you wish to delete by moving to it with the Up- or Down-arrow key. Indicate your selection by placing a check mark before the selected name(s) with the space bar.

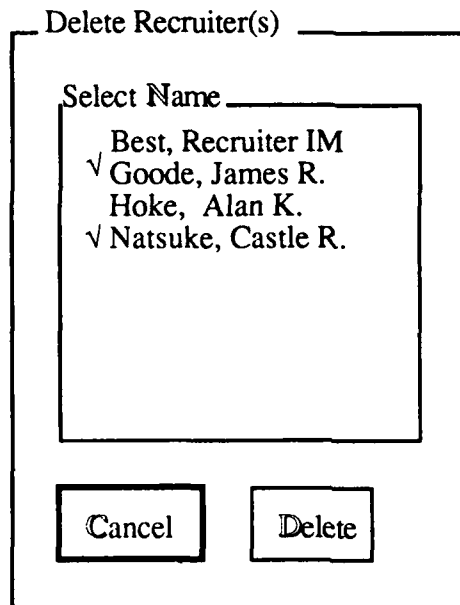


Figure 5.2.5. Delete Recruiter Dialog

Having selected the recruiter's(s') name(s), TAB to the "Delete" radio button and press the Enter key to delete the selected recruiter(s) or use the hot-key Alt-D to delete the recruiter(s).

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the selection.

#### 5.2.6. Quit Dialog

When a recruiter is finished using the system, the Quit dialog (Figure 5.2.6.) is provided to effect an orderly exit. This dialog is displayed with the Cancel option as the default option. The recruiter must press the Tab key to move to the Quit option and then press the Enter key to quit. This dialog was defined in this manner to preclude accidental exit by a recruiter.

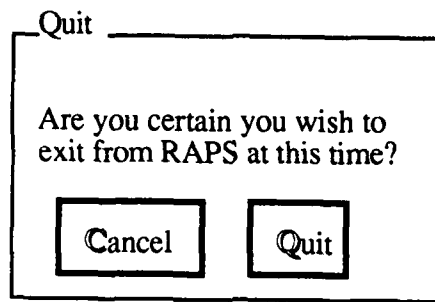


Figure 5.2.6(a). Quit Dialog

### 5.2.7. Save Database Dialog

Before the recruiter is allowed to exit RAPS, he is asked whether or not he wishes to save changes to the Database (Figure 5.2.6(b).). This dialog accepts only a "Y" for yes or an "N" for a No response. Responding with "Y" will save all database changes made; responding with an "N" will discard all changes made.

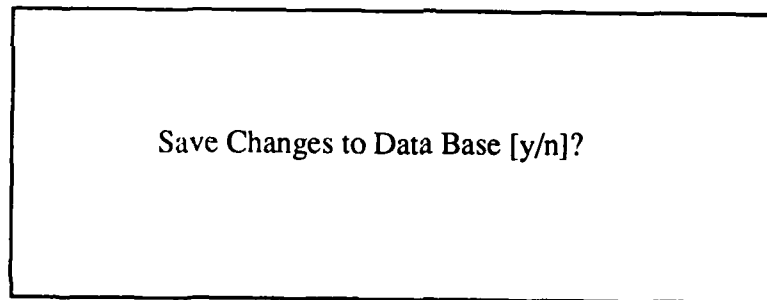


Figure 5.2.6(b). Save Database Dialog

### 5.3. Lead-Generation Phase

A recruiter builds and maintains lists of leads from many sources, such as school class lists, REACT lists, ASVAB lists, and job service lists. The Lead Function on the Menu Bar accesses six Lead dialogs: (1) Select, (2) Update, (3) Add, (4) Import, (5) Delete, and (6) Print (Figure 5.3.). Any of these dialogs may be accessed by using the arrow keys to highlight the option of choice and then pressing the Enter key. Alternatively, the hot-key associated with each option may be pressed.

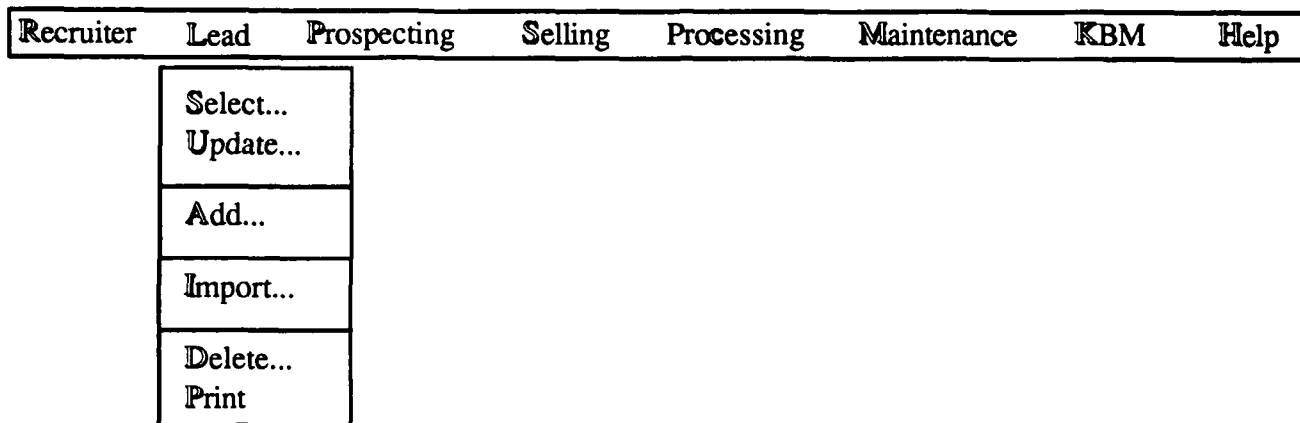


Figure 5.3. Lead Pull-Down Menu

"Select" presents a list of the currently documented leads and allows the user to identify and access the particular lead of interest. "Update" allows changing information about a lead or adding information which was missing when the lead was initially entered into the system. "Add" allows the addition of new leads to the Lead database. Leads are added by providing any information currently available about the lead: name, address, phone number, and the source of the lead. "Import" allows leads to be added from electronic media; a file of leads can be directly imported to RAPS. "Delete" allows deletion of a lead(s) from the Lead database. "Print" creates paper copies of the entire list of leads or of selected leads.

#### 5.3.1. Select Lead Dialog

The Select Lead dialog (Figure 5.3.1.) presents a radio list box in which only one lead out of many may be selected. Find the name of the lead you wish to access in the box and select it by moving to it with the Up- or Down-arrow key. The name selected is marked by an arrowhead.

Having selected the lead, TAB to the "Select" radio button. Pressing the Enter key at this point will access the data relevant to the selected lead. The hot-key Alt-S may also be used to select the designated lead.

Pressing the hot-key Alt-C will cancel the selection. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the selection.

Select Lead

Select Name

- Best, Lead IM
- Goodlead, John R.
- Hoyle, Arnold A.
- Nash, Jackson T.

Cancel Select

Figure 5.3.1. Select Lead Dialog

### 5.3.2. Update Lead Dialog

The Update Lead dialog (Figure 5.3.2.) presents all data currently relevant to the selected lead. TAB to the Address box to enter or modify address data. TAB to the Phone box to enter or modify the lead's telephone number. Next, TAB to the radio list box labeled "Source" to identify the source of the lead. Use the up and down arrow keys to enter the appropriate lead source in the box. If necessary, use the edit box labelled "Other Source" to identify other sources for the lead data. Finally, TAB to the date box to enter the date when the lead was identified.



Having updated lead information, TAB to the "Update" radio button. Pressing the Enter key at this point will update the lead data with the changes made. The hot-key Alt-U may also be used to update the lead data with the changes made.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

Update Lead

<b>Name</b> Goodlead, John R.	<b>Source</b> REACT List
<b>Address</b> Street City, State ZIP 00001:001 O	<b>Other Source</b> Source of other lead 00001:001
<b>Phone</b> (aaa)/ppp-nnnn	<b>Identified</b> mm/dd/yy
<b>Cancel</b>	<b>Update</b>

Figure 5.3.2. Update Lead(s) Dialog

### 5.3.3. Add Lead Dialog

The Add Lead dialog (Figure 5.3.3.) allows new leads to be added to the Lead database. Enter the name of the new lead in the Name box. TAB to the Address box to enter address data. TAB to the Phone box to enter the lead's telephone number. TAB to the radio list box under "Source" to identify the source of the lead. TAB to the edit box labeled "Other Source" to identify other sources for the lead data. Finally, TAB to the date box to enter the date when the lead was identified.

Having added all available lead information, TAB to the "Add" radio button. Pressing the Enter key at this point will add the lead data just entered. The hot-key Alt-A may also be used to add the lead data just entered.

Pressing the hot-key Alt-C will cancel the operation. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the operation of adding the lead.

Add Lead

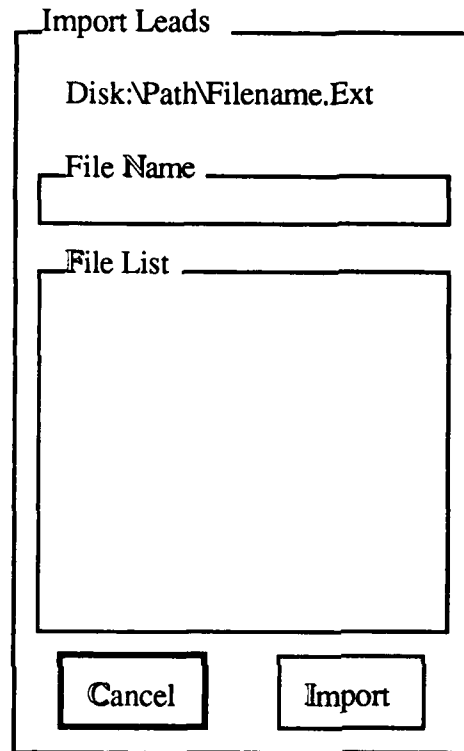
Name Last, First, M.	Source REACT List
Address Street City, State ZIP 00001:001 O	Other Source Source of other lead 00001:001
Phone (aaa)/ppp-nnnn	Identified: mm/dd/yy
Cancel	Add

Figure 5.3.3. Add Lead(s) Dialog

#### 5.3.4. Import Lead Dialog

The Import Lead dialog (Figure 5.3.4.) allows new leads to be added to the Lead database from external, electronic sources such as ASVAB lists, REACT lists, electronic high school lists, etc. Copy the file to be imported onto a disk. Then invoke RAPS and log the recruiter for whom the leads are intended. Next, access the Import Lead dialog. Enter the filename to be accessed into the field labeled "File Name" using the format shown above. If the filename is not remembered, TAB or use the Alt-F hot-key to advance to the radio list of filenames. Use the Up- and Down-arrow keys to highlight the filename to be imported. When the filename is entered, or is selected in the radio list, TAB to the Import radio button and push the Enter key. Alternatively, press the Alt-I radio button to import all of the leads.

Pressing the hot-key Alt-C will cancel the operation. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the operation of importing leads.



The dialog box is titled "Import Leads". It contains a text field with the placeholder "Disk:\Path\Filename.Ext". Below this is a "File Name" label followed by an empty text input box. Underneath is a "File List" label followed by a large empty rectangular area, likely for a list of files. At the bottom, there are two buttons: "Cancel" on the left and "Import" on the right.

Figure 5.3.4. Import Leads Dialog

At present, the electronic file to be imported must be in comma-delimited, carriage return terminated form. Each file must have 4 fields. The first field contains the name of the lead, the second field contains the first address line, the third field contains the second address line, and the fourth field contains the telephone number of the lead.

### 5.3.5. Delete Lead(s) Dialog

The Delete Lead(s) dialog (Figure 5.3.5.) presents a choice list box in which one or more leads may be selected. Find the lead(s) you wish to delete from the database by moving to them with the Up- or Down-arrow key. Indicate your selection by placing a check mark before the selected name(s) with the space bar.

Having selected the lead name(s), TAB to the "Delete" radio button and press the Enter key to delete the selected lead(s) from the Lead database. You may also use the hot-key Alt-D to delete the lead(s).

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the deletion.

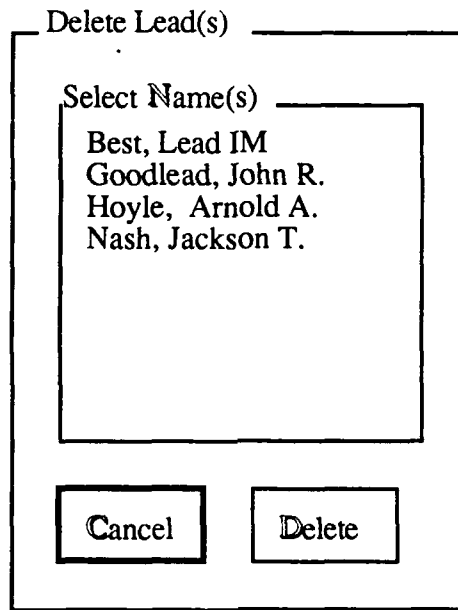


Figure 5.3.5. Delete Lead(s) Dialog

#### 5.3.6. Print Lead(s) Dialog

The Print Lead(s) dialog (Figure 5.3.6.) allows the printing of leads. The Print Group box allows you to identify whether to print the entire group of leads or only selected leads.

Print Lead(s)

Print Group <input type="text" value="All"/>	Select Name(s) Best, Lead IM Goodlead, John R. Hoyle, Arnold A. Nash, Jackson T.
Print Order <input type="text" value="by Name"/>	
<input type="button" value="Cancel"/> <input type="button" value="Print"/>	

Figure 5.3.6. Print Lead(s) Dialog

TAB to the Print Order box to indicate whether to print leads in alphabetic order by name, in alphabetic order by source, or in order of priority.

To select specific leads to print, TAB to the Select Name(s) box and select the leads to print by moving to them with the Up or Down arrow key. Select (or deselect) each name with the space bar. The selected names will be marked by a check mark.

Having selected the leads, TAB to the "Print" radio button and press the Enter key to print the selected leads from the Lead database. You may also use the hot-key Alt-P to print the selected leads.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the print. Leads can be printed in a format similar to the Lead Refinement List (USAREC Form 539). This format allows the printed form to be used for lead refinement in the Prospecting Phase .

## 5.4. Prospecting Phase

The Prospecting Phase involves recruiter contact with identified leads and the updating of lead information as it is obtained. The Prospecting Phase continues until such time as the lead agrees to a sales interview, at which time the lead becomes a prospect, or it is determined there is no reason to pursue this lead. Prospecting is implemented in RAPS through the Prospecting menu.

Recruiter	Lead	Prospecting	Selling	Processing	Maintenance	KBM	Help
		Select...					
		Priority...					
		Contacts...					
		Disposition...					
		Delete...					
		Print...					

Figure 5.4. Prospecting Menu

The Prospecting Menu (Figure 5.4.) provides six options. The "Select" option provides access to a dialog for selecting a lead for update. The "Priority" option accesses a dialog that allows updating the lead's priority based on ASVAB qualifications, career decisions, and AFQT score range. The "Contacts" option maintains information regarding recruiter contacts with the lead. The "Disposition" option records the final status of the lead. The "Delete" option allows one or more leads to be deleted. The "Print" option allows the printing of records for selected leads.

To access the Prospecting options, use the Up- and Down-arrow keys to move to the desired option, then press the Enter key. Alternatively, use the hot-key identified on the Pull-Down Menu to activate the desired option.

### 5.4.1. Select Lead Dialog

The Select Lead dialog of the Prospecting Phase presents a radio list box in which only one lead out of many may be selected(Figure 5.4.1(a).). Find the name of the lead you wish to access in the box and select it by moving to it with the Up- or Down-arrow key. The name selected is marked by an arrowhead.

Select Lead

Select Name

- Best, Lead IM
- Goodlead, John R.
- Hoyle, Arnold A.
- Nash, Jackson T.

Cancel

Dial

Update

Select

Figure 5.4.1(a). Select Lead Dialog

Having selected the lead, use TAB or any of the defined hot-keys to move to one of the four radio: "Cancel", "Dial", "Update", or "Select". "Cancel" cancels the selection and returns to the Prospecting menu. "Dial" automatically passes the lead phone number to the RAPS modem and dials the lead's telephone number. A message is displayed on the screen to report the dialing function and will disappear when (1) the telephone is answered, (2) a busy signal or no-answer occurs, or (3) the user presses a key.

Dialing: Hoyle, Arnold A.

at: (703) 555-1212

Figure 5.4.1(b). Telephone Dialing Message

The "Update" radio button Alt-U accesses the Prospect Lead Contact dialog and allows the user to complete information about the telephone contact initiated through RAPS.

The "Select" radio button allows access to data relevant to the selected lead. The hot-key Alt-S may also be used to select the designated lead.

5.4.2. Update Lead Priority Dialog

RAPS maintains lead records in terms of priorities established by USAREC policies and regulations as well as by Mission Box. Priorities depend upon applicant status, and these can be set in the Lead Priority dialog box. The lead name defaults to the name selected earlier. The date of update defaults to the system date.

Select the most appropriate radio button under Lead Priority (Figure 5.4.2.) by using the Up- or down-arrow key.

Lead Priority

Priority:	Career:	AFQT Range:
(*) ASVAB Taken	( ) Military	( ) 50-99
( ) No ASVAB	(*) Undecided	(*) 31-49
( ) Not Qualified (DEP)	( ) Work	( ) 26-38
( ) Not Qualified (601-210)	( ) College/VOTECH	( ) 16-25

Cancel Update

Figure 5.4.2. Update Lead Priority Dialog



Next, TAB to the radio buttons that can be used to indicate the applicant's career choice. Again, use the Up- or Down-arrow key to select the most appropriate radio button.

Now, TAB to the radio button that describes the applicant's AFQT score. Use the Up- or Down-arrow key to select the most appropriate button.

TAB to the "Update" button and push the Enter key to record your selections in the Lead database. Alternatively, you may use the hot-key Alt-U to update the record.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

The Update Lead Priority dialog allows the recruiter to indicate several items used in determining the lead priority. The RAPS system uses the individual entries in this dialog to assign lead-processing priority. This priority may then be used to print a priority-ordered list for prospecting.

#### **5.4.3. Update Lead Contacts Dialog**

RAPS maintains records of lead contacts (Figure 5.4.3.). Use the Attempt box to enter the date and time you attempted to contact the selected lead. RAPS attempts to use the number of contact attempts and successes to reappraise the priorities established during the Update Lead Priority dialog.

Multiple contacts are entered, edited, and deleted through this dialog box. Upon first entry to the Lead Contacts dialog, data relevant to the most recent contact are presented in the box. If there are no data, you can enter data and use the "Add" radio button to insert these data into the applicant's data record.

Lead Contacts

Attempt:	mm/dd/yy; hh:mm	Comments          00001:001
Code:	HC - House Call	
<input type="button" value="Cancel"/> <input type="button" value="Add"/> <input type="button" value="Next"/> <input type="button" value="Prev"/> <input type="button" value="Delete"/> <input type="button" value="Update"/>		

Figure 5.4.3.. Update Lead Contacts Dialog

TAB to the Code box to indicate which kind of contact was attempted: house call, mailing, no answer to a phone call, not at home during a visit, or a successful phone call. Use the Up- or Down-arrow key to select the most appropriate contact attempt.

TAB to the Comments edit box to enter your comments regarding the contact. This box is entered in overstrike mode. Terminate your lines with a carriage return so that all of the text remains visible in the box. If necessary, you may toggle between overstrike mode and insert mode using the Ins key. To exit the edit box and advance to the next field, use the Tab key. Alternatively, you may use any hot-key (e.g., Alt-U) to advance to the next field of interest.

TAB to the "Add" radio button and push the Enter key to record your selections in the Lead database if you are entering data for a new lead contact. You may also use the hot-key Alt-A to add new contact information.

TAB to the "Update" radio button and push the Enter key to record your selections in the Lead database if you are updating a previous entry. Alternatively, you may use the hot-key Alt-U to update the record.

Pressing the hot-key Alt-C will cancel your work. Pressing the Enter key while the "Cancel" radio button is selected will also cancel your entries.

To access data from earlier contacts, TAB to the "Prev" radio button and press the Enter key. Alternatively, press the hot-key Alt-V. Each use of the "Prev" radio button will access data from earlier contacts.

To access data from later contacts, TAB to the "Next" radio button and press the Enter key. Alternatively, press the hot-key Alt-N. Each use of the "Next" radio button will access data from later contacts.

To modify and update information relevant to any contact, use the "Prev" or "Next" radio button to find the contact, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press the Enter key. Alternatively, press the hot-key Alt-U to update the record.

To delete information relevant to any contact, use the "Prev" or "Next" radio button to find the contact, TAB to the "Delete" radio button, and press the Enter key. Alternatively, press the hot-key Alt-D to delete the data.

#### 5.4.4. Update Lead Disposition Dialog

RAPS maintains records of what happens to leads so that it can continually reassess lead priority. Use the "Disposition" radio button to indicate lead disposition. Use the Up- or Down-arrow key to select the most appropriate disposition from the list (figure 5.4.4.).

Disposition

Disposition:

- (\*) Unwilling to commit
- ( ) Lead becomes prospect
- ( ) Found to be disqualified
- ( ) Moved out of zone
- ( ) Referred to another
- ( ) Unable to contact
- ( ) Enlisted in another service
- ( ) Totally without interest
- ( ) Plans on continuing education
- ( ) Attending college or other school
- ( ) Enlist into U.S. Army
- ( ) Other (See Comments)

Comp:

Comments

Cancel

Update

Figure 5.4.4. Update Lead Disposition Dialog

TAB to the Comments edit box to enter your comments regarding the contact. The Comments box is entered in overstrike mode. To toggle between insert mode and overstrike mode, use the Ins key. To exit the edit box, use the Tab key or Shift+Tab to advance to the next or previous control. Alternatively, use the hot-key Alt-U to advance to the "Update" button. Because the edit box is scrollable, it may be useful to insert carriage returns into the box so that subsequent retrieval will format the text into the edit box.

TAB to the "Update" button and push the Enter key to record your selections in the lead database. Alternatively, use the hot-key Alt-U to update the record.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

#### 5.4.5. Delete Lead(s) Dialog

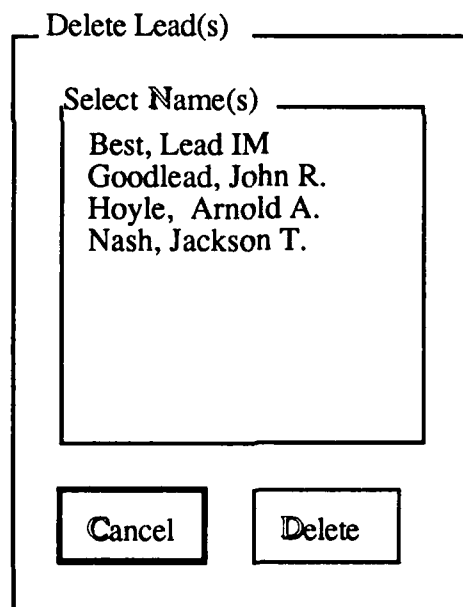


Figure 5.4.5. Delete Lead(s) Dialog

The Delete Lead(s) dialog (Figure 5.4.5.) presents a choice list box in which any number of leads may be selected. Find the leads you wish to delete from the database by moving to them with the Up- or Down-arrow key. Pressing

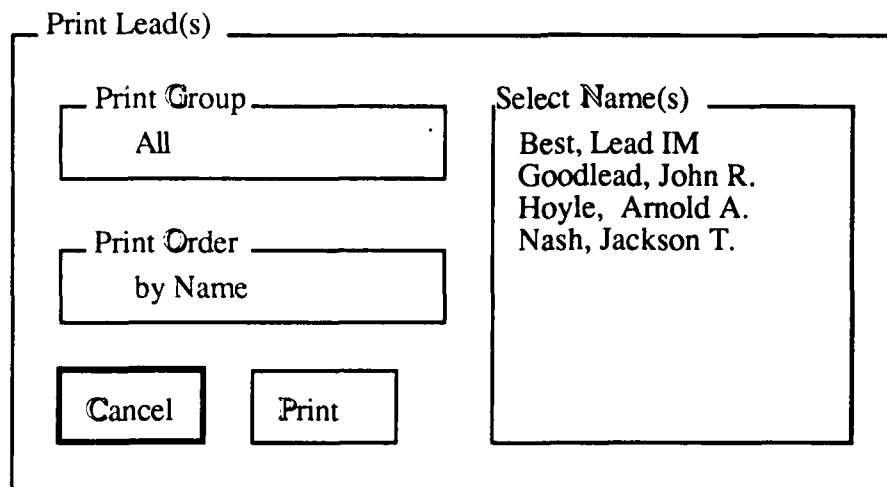
the space bar will toggle a check mark in front of the lead name, marking or unmarking the name for deletion.

Having selected the leads' names, TAB to the "Delete" radio button and press the Enter key to delete the selected leads from the Lead database. You may also use the hot-key Alt-D to delete the leads.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the deletion.

#### 5.4.6. Print Lead(s) Dialog

The Print Lead(s) dialog (Figure 5.4.6.) allows the printing of leads. The Print Group box allows you to identify whether to print the entire group of leads or only selected leads. Leads will be printed in a format similar to the Lead Refinement List (USAREC Form 539). This format allows the printed form to be used for lead refinement in the Prospecting Phase.



The dialog box is titled "Print Lead(s)". It contains three main sections. On the left, there are two stacked boxes: "Print Group" with "All" selected, and "Print Order" with "by Name" selected. Below these are two buttons: "Cancel" and "Print". On the right, there is a box titled "Select Name(s)" containing a list of names: "Best, Lead IM", "Goodlead, John R.", "Hoyle, Arnold A.", and "Nash, Jackson T.".

Figure 5.4.6. Print Lead(s) Dialog

TAB to the Print Order box to indicate whether to print leads in alphabetic order by name, in alphabetic order by source, or in alphabetic order by priority.

To select specific leads to print, TAB to the Select Name(s) box and select the leads to print by moving to them with the Up- or Down-arrow key. Select (or deselect) each name with the space bar. The selected names will be marked by a check mark.

Having selected the leads, TAB to the "Print" radio button and press the Enter key to print the selected leads from the Lead database. You may also use the hot-key Alt-P to print the selected leads.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the print.

## 5.5. Selling Phase

The Selling Phase begins when an individual agrees to a sales interview. At this time the lead becomes a prospect and RAPS creates a prospect record. The prospect record is very similar to the Prospect Data Record (USAREC Form 200) used for manual processing.

The individual remains a prospect until such time as an agreement to process is reached or it is determined that the individual should no longer be considered a prospect. Upon agreement to process, RAPS creates an Applicant Data Record very similar in content to the Department of the Army Form 1966. At this time further processing is performed under the Processing Phase dialogs within the RAPS system.

Recruiter	Lead	Prospecting	Selling	Processing	Maintenance	KBM	Help
-----------	------	-------------	---------	------------	-------------	-----	------

Select...

Prospect Data...  
Prequalification...  
Citizenship Data...  
Prior Service...  
Law Violations...  
Personal Data...  
Test Data...  
Test Data/Physical...  
Interview Remarks...

Disposition

Delete...  
Print...

MS-DOS Access

Figure 5.5. Selling Menu

The Selling Menu (Figure 5.5.) provides options for selecting a prospect and then updating information relating to that prospect. Options are also provided for deleting and printing prospects. To access any of the options, use the Up- or Down-arrow key to highlight the desired option and then press

the Enter key. Alternatively, use the defined hot-key to launch the option directly.

### 5.5.1. Select Prospect Dialog

The Select Prospect dialog of the Selling Phase (Figure 5.5.1.) presents a radio list box in which only one prospect out of many may be selected. Find the name of the prospect you wish to access in the box and select it by moving to it with the Up- or Down-arrow key. The name selected is marked by an arrowhead.

Select Lead

Select Name

- Best, Lead IM
- Goodlead, John R.
- Hoyle, Arnold A.
- Nash, Jackson T.

Cancel Select

Figure 5.5.1. Select Prospect Dialog

Having selected the prospect, TAB to the "Select" radio button. Pressing the Enter key at this point will access the data relevant to the selected prospect. The hot-key Alt-S may also be used to select the designated prospect.

Pressing the hot-key Alt-C will cancel the selection. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the selection.

### 5.5.2. Update Prospect Data Dialog



The Update Prospect Data dialog (Figure 5.5.2.) provides for entry or update of basic information about the prospect.

Prospect Data

Legal Name \_\_\_\_\_  
Last, First M.

Contact \_\_\_\_\_  
mm/dd/yy

Follow-up \_\_\_\_\_  
mm/dd/yy

Address \_\_\_\_\_  
Street  
City, State ZIP  
00001:001

Alias \_\_\_\_\_  
Last, First M.

Phone \_\_\_\_\_  
(aaa)/ppp-nnnn

Interview \_\_\_\_\_  
mm/dd/yy

Location \_\_\_\_\_  
City, State

Cancel Update

Figure 5.5.2. Update Prospect Dialog

In the name field, fill in the last name, first name, middle initial, and qualifier (Jr, III, etc.). TAB to the Address field. The Address edit box is entered in overstrike mode. To toggle between insert mode and overstrike mode, use the Ins key. Two numbers, separated by a colon, appear at the bottom of the box. The first number indicates the line number being edited; the second number indicates the column. An "I" or an "O" appears when toggling the Ins key to indicate whether editing is implemented in insert mode or overstrike mode. The Address box will accommodate any number of address lines.

To exit the Address box and advance to the next field, use the Tab key; to return to the previous field, use Shift+TAB. Alternatively, use the appropriate hot-key to advance to the next field of interest.

Use the Phone field to enter the prospect's telephone number.

Use the Contact field to enter the contact date.

Use the Follow-up field to enter the date when a follow-up contact should be made. This allows RAPS to prompt the recruiter with reminders on the appropriate days.

Use the Alias field to enter any known aliases used by the prospect.

Use the Lead Source field to indicate the source of the prospect. This field is a radio list box. Scroll through the Alternatives with the Up- and Down-arrow keys until the appropriate source appears in the source window. Then TAB to the next field.

Use the Interview field to enter the sales interview date arranged for the prospect.

Use the Location field to enter the location.

Having entered the relevant data, TAB to the "Update" radio button. Pressing the Enter key at this point will enter the updated information into the database. The hot-key AltU) may also be used to update the information.

Pressing the hot-key AltC will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

### **5.5.3. Update Prospect Prequalification Data Dialog**

The Update Prospect Prequalifications Data dialog (Figure 5.5.3.) allows entry or update of prequalification data. If any entries are made that indicate disqualification of the prospect, a pop-up dialog will be presented. This dialog will provide disposition options.

The Health edit box is used to enter text as needed to describe the applicant's health. This box is entered in overstrike mode. To toggle between insert mode and overstrike mode, use the Ins key. Two numbers, separated by a colon, appear at the bottom of the box. The first number indicates the line number being edited; the second number indicates the column. An "I" or an "O" appears when toggling the Ins key to indicate whether editing is implemented in insert mode or overstrike mode. The Health box will accommodate any number of lines.

To exit the Health box and advance to the next field, use the Tab key; Shift+Tab returns to the previous field. Alternatively, use the appropriate hot-key to advance to the next field of interest.

Prequalification Data

Name: Thomas, Gerald R.		Birth Date: mm/dd/yy	
Health 00001:001		Birth Place 00001:001	
Last School Attended 00001:001		Sex Male Female	Citizenship U.S. Other
Education Code 01-NHSG	Ht [ ]	Wt [ ]	M Status M - Married
			No Dep 0
Cancel		Update	

Figure 5.5.3. Update Prospect Prequalification Data Dialog

The Last School Attended is an edit box like the Health box, and is entered, exited, and toggled like the Health box. Use this box to enter information regarding the last school attended by the prospect.

Enter the Education Code by scrolling through the radio list using the Up- or Down-arrow key until the correct education code appears in the window. Then continue to the next field by using the Tab key, Shift+Tab, or a hot-key.

Enter the prospect's birth date in the corresponding field.

The Birth Place is an edit box like the Health box and follows the same editing conventions. Use it to enter the applicant's place of birth.

Identify the applicant's sex by scrolling to Male or Female using the Up- or Down-arrow key.

Identify the applicant's citizenship by toggling to U.S. or Other using the Up- or Down-arrow key.

Enter the prospect's height in inches and weight in pounds.

Identify the applicant's marital status by toggling to the appropriate identifier using the Up- or Down-arrow key.

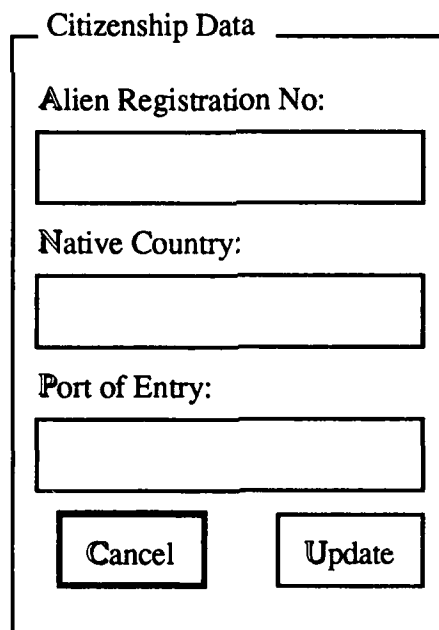
Enter the prospect's number of dependents.

Having entered the relevant data, TAB to the "Update" radio button. Pressing the Enter key at this point will enter the updated information into the database. The hot-key Alt-U may also be used to update the information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

#### **5.5.4. Update Prospect Citizenship Data Dialog**

The Update Prospect Citizenship Data dialog (Figure 5.5.4.) is presented for data entry or update if the recruiter specifies the prospect is an alien on the Update Prospect Prequalification Data dialog.



Citizenship Data

Alien Registration No:

Native Country:

Port of Entry:

Cancel Update

Figure 5.5.4. Update Prospect Citizenship Data Dialog

If the prospect is not a U.S. citizen, it is necessary to complete additional citizenship information.

Enter the prospect's alien registration number and TAB to the next field to enter the prospect's native country. TAB again to advance to the Port of Entry field.

Having entered the relevant data, TAB to the "Update" radio button. Pressing the Enter key at this point will enter the updated information into the database. The hot-key Alt-U may also be used to update the information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

#### 5.5.5. Update Prior Service Data Dialog

The Prior Service Data dialog (Figure 5.5.5.) is used to enter prior service information. TAB through the fields to complete all of the prior service information. Use the defined hot-keys to jump from field to field.

Prior Service Data

RE/SPN: refer 214 Form	Grade N/A	MOS 
Service N/A	Months 	Separated mm/dd/yy
Reserve Information		
Unit 	Component N/A	DOS 
Status 	Cancel	Update

Figure 5.5.5. Update Prior Service Data Dialog

Having entered the relevant data, TAB to the "Update" radio button. Pressing the Enter key at this point will enter the updated information into the database. The hot-key Alt-U may also be used to update the information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

### 5.5.6. Update Law Violations Data Dialog

RAPS maintains all data relevant to law violations. Use this dialog box (Figure 5.5.6.) for that purpose.

Multiple offenses are entered, edited, and deleted through this dialog box. Upon first entry to the Law Violations Data dialog, data relevant to the most recent offense are presented in the box. If there are no data, you can enter data and use the "Add" radio button to insert these data in the applicant's data record.

Law Violations

Offense	Category	Off Date
	Minor Non	mm/dd/yy
Place	Age	Disposition
City, State		
Court	City, State	
Location of Check	Date of Check	
City, State	mm/dd/yy	
Cancel	Add	Prev
Next	Delete	Update

Figure 5.5.6. Update Law Violations Dialog

To access data for earlier offenses, TAB to the "Prev" radio button and press the Enter key. Alternatively, press the hot-key Alt-V. Each use of the "Prev" radio button will access data for earlier offenses.

To access data for later offenses, TAB to the "Next" radio button and press the Enter key. Alternatively, press the hot-key Alt-N. Each use of the "Next" radio button will access data for later offenses.

To modify and update information relevant to any offense, use the "Prev" or "Next" radio button to find the offense, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press the Enter key. Alternatively, press the hot-key Alt-U to update the record.

To delete information relevant to any offense, use the "Prev" or "Next" radio button to find the offense, TAB to the "Delete" radio button, and press the Enter key. Alternatively, press the hot-key Alt-D to delete the data.

To enter data, begin with the most recent offense. Enter the offense description, offense category (misdemeanor, felony), offense date, place of offense, age of applicant at time of offense, disposition of case, court, location of the police station where the law violation was checked, and the date of the check. Next, TAB to the "Add" radio button and press the Enter key to "Insert" these data to the applicant's record. Alternatively, you may press the hot-key Alt-I to insert the data.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel your work.

#### **5.5.7. Update Personal Data Dialog**

Use the Personal Data dialog box (Figure 5.5.7.) to enter personal data regarding the prospect. The "Name" and "Date of update" default to the selected name and today's date. The rest of the fields may be TABbed through or the appropriate hot-keys may be used to directly access each field of importance.

Personal Data

SSN 999-99-99999	Hair	Eyes
Race	Religion	
Drivers License No	State	Exp. Date mm/dd/yy
Cancel		Update

Figure 5.5.7. Update Personal Data Dialog

Having entered the relevant data, TAB to the "Update" radio button. Pressing the Enter key at this point will enter the updated information into the database. The hot-key Alt-U may also be used to update the information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

### 5.5.8. Test Data Dialog

RAPS maintains applicant test data (Figure 5.5.8.). The first field, "Test Type," is used to indicate whether the test administered was the CAST or EST. Select the test type with the Up- and Down-arrow key(s). Use the Tab key or a hot-key to advance to the next field.



Test Data

Test Type CAST	Ver. <input type="text"/>	Score <input type="text"/>	Date mm/dd/yy <input type="text"/>	AFQT <input type="text"/>	DAFQT <input type="text"/>
-------------------	------------------------------	-------------------------------	--	------------------------------	-------------------------------

Special Scores:

WOFT <input type="text"/>	OCS <input type="text"/>	DLAB <input type="text"/>	DLAT <input type="text"/>	MVDB <input type="text"/>
------------------------------	-----------------------------	------------------------------	------------------------------	------------------------------

Figure 5.5.8. Update Test Data Dialog

Enter the applicant's score in the Score field, and the date of test administration in the Date field. Enter the version code in the field labeled "Ver." Enter the AFQT and DAFQT test scores as appropriate. Also fill in the component scores and special scores as appropriate.

Having entered the relevant data, TAB to the "Update" radio button. Pressing the Enter key at this point will enter the updated information into the database. The hot-key Alt-U may also be used to update the information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

### 5.5.9. Test Data/Physical Dialog

RAPS maintains applicant physical test data (Figure 5.5.9.). Fill in the appropriate 'Physical Profile'. Use TAB or a hot-key to advance to the next field and fill in the "Color Perception". Continue to use the Tab key or a hot-key to complete the rest of the physical data.

Having entered the relevant data, TAB to the "Update" radio button. Pressing the Enter key at this point will enter the updated information into the database. The hot-key Alt-U may also be used to update the information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

Test Data/Physical

Physical Profile

P  U  L  H  E  S  X

Color Perception  Ht  Wt  Date

mm/dd/yy

Figure 5.5.9. Test Data/Physical Dialog

#### 5.5.10. Interview Remarks Dialog

Interview Remarks

Remarks

00001:001

Figure 5.5.10. Interview Remarks Dialog

This dialog box (Figure 5.5.10.) allows you to enter remarks pertinent to the selling interview. The Remarks edit box is used for this purpose; it is entered with the cursor in block form. While the cursor is in block form, the Tab key will allow the Remarks box to be skipped. Any other key will enter the Remarks box (the cursor will change to underscore form), and editing will be implemented in insert mode. To toggle between insert mode and overstrike mode, use the Ins key. Two numbers, separated by a colon appear at the bottom of the box. The first number indicates the line number being edited; the second number indicates the column. An "I" or an "O" appears when toggling the Ins key to indicate whether editing is implemented in insert mode or overstrike mode. The Remarks box will accommodate any number of lines.

To exit the Remarks box, use Ctrl-Q to change the cursor back to block form and enable you to TAB to one of the two radio buttons at the bottom of the dialog box. Alternatively, use a hot-key to Cancel Alt-C or to Update Alt-U.

Having TABbed to the "Update" radio button, pressing the Enter key will enter the updated information into the database. The hot-key Alt-U may also be used to update the information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

### **5.5.11. Disposition Dialog**

RAPS maintains records of what happens to leads so that it can continually assess and update lead priority. Use the "Disposition" radio buttons to indicate lead disposition. Use the Up- or Down-arrow key to select the most appropriate disposition from the list (Figure 5.5.11.).

Disposition

<p>Disposition:</p> <ul style="list-style-type: none"> <li>(*) Unwilling to commit</li> <li>( ) Lead becomes prospect</li> <li>( ) Found to be disqualified</li> <li>( ) Moved out of zone</li> <li>( ) Referred to another</li> <li>( ) Unable to contact</li> <li>( ) Enlisted in another service</li> <li>( ) Totally without interest</li> <li>( ) Plans on continuing education</li> <li>( ) Attending college or other school</li> <li>( ) Enlist into U.S. Army</li> <li>( ) Other (See Comments)</li> </ul>	<p>Comp: <input type="text" value="RA"/></p> <p>Comments <input type="text"/></p> <p>00001:001</p>
<input type="button" value="Cancel"/>	<input type="button" value="Update"/>

Figure 5.5.11. Disposition Dialog

TAB to the Comments edit box to enter your comments regarding the contact. The Comments box is entered in overstrike mode. To toggle between insert mode and overstrike mode, use the Ins key. To exit the edit box, use the Tab key or Shift+Tab to go to the next or previous control. Alternatively, use the hot-key Alt-U to advance to the "Update" button.

TAB to the "Update" button and push the Enter key to record your selections in the Lead database. Alternatively, use the hot-key Alt-U to update the record.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

### 5.5.12. Delete Prospect(s) Dialog

The Delete Prospect(s) dialog (Figure 5.5.12.) provides for the deletion of one or more prospects from the database.

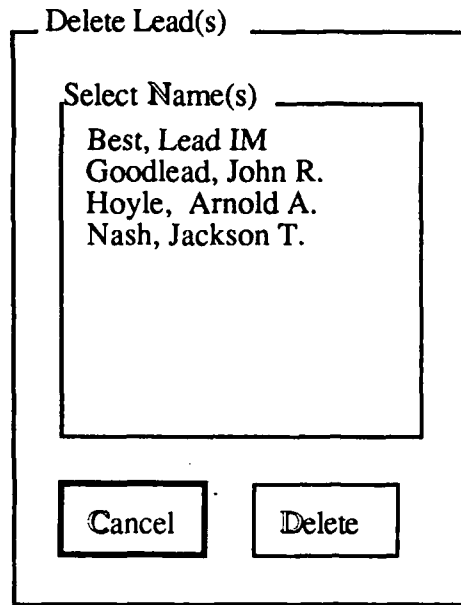


Figure 5.5.12. Delete Prospect(s) Dialog

The Delete Prospect(s) dialog presents a choice list box in which one or more prospects may be selected for deletion. Find the prospects you wish to delete from the database by moving to them with the Up- or Down-arrow key. Select or deselect each prospect with the space bar. Prospects selected for deletion will be marked with a check mark.

Having selected the prospect's(s') name(s), TAB to the "Delete" radio button and press the Enter key to delete the selected prospect(s) from the prospect database. You may also use the hot-key Alt-D to delete the prospect(s).

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the deletion.

### 5.5.13. Print Prospect(s) Dialog

The Print Prospect(s) dialog (Figure 5.5.13.) allows printing of one or more sets of prospect information. The recruiter may elect to print all prospects in order of name, source or priority, or the recruiter may select a set of prospects to print.

Print Lead(s) \_\_\_\_\_

Print Group _____ <div>All</div>	Select Name(s) _____ Best, Lead IM Goodlead, John R. Hoyle, Arnold A. Nash, Jackson T.
Print Order _____ <div>by Name</div>	
<div>Cancel</div> <div>Print</div>	

Figure 5.5.13. Print Prospect(s) Dialog

The Print Group box allows you to identify whether to print the entire group of prospects or only selected prospects. Use the Up- or Down-arrow key to indicate which printing option you want.

Next, TAB to the Print Order box to indicate whether to print leads in alphabetic order by name, in alphabetic order by source, or in alphabetic order by priority. Use the Up- or Down-arrow key to indicate your choice.

To select specific leads to print, TAB to the Select Name(s) box and select the leads to print by moving to them with the Up- or down- arrow key. Select (or deselect) each name with the space bar. The selected names will be marked by a check mark.

Having selected the leads, use TAB to move to the "Print" radio button and press the Enter key to print the selected leads from the lead database. You may also use the hot-key Alt-P to print the selected leads.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the print.

#### 5.5.14. MS-DOS Access

Accessing MS-DOS through RAPS will allow you to run any other available MS-DOS software, including JOIN. To access MS-DOS, move to the MS-DOS Access option in the Selling Menu with the Up- or Down-arrow key and

press the Enter key. Alternatively, use the hot-key Alt-M to directly jump to DOS.

Upon entry to DOS, you will be presented with the familiar prompt:

A:>

At that point, you can run any MS-DOS-compatible software, including JOIN. Upon exiting the software to the DOS prompt (A:>), you can type "Exit" to return to RAPS.

## 5.6. Processing Phase

The Processing Phase begins following a prospect's agreement to process. At this time the prospect becomes an applicant and an Applicant Data Record (ADR) is created. RAPS processing encompasses all activities related to qualifying and preparing the applicant for the oath of enlistment.

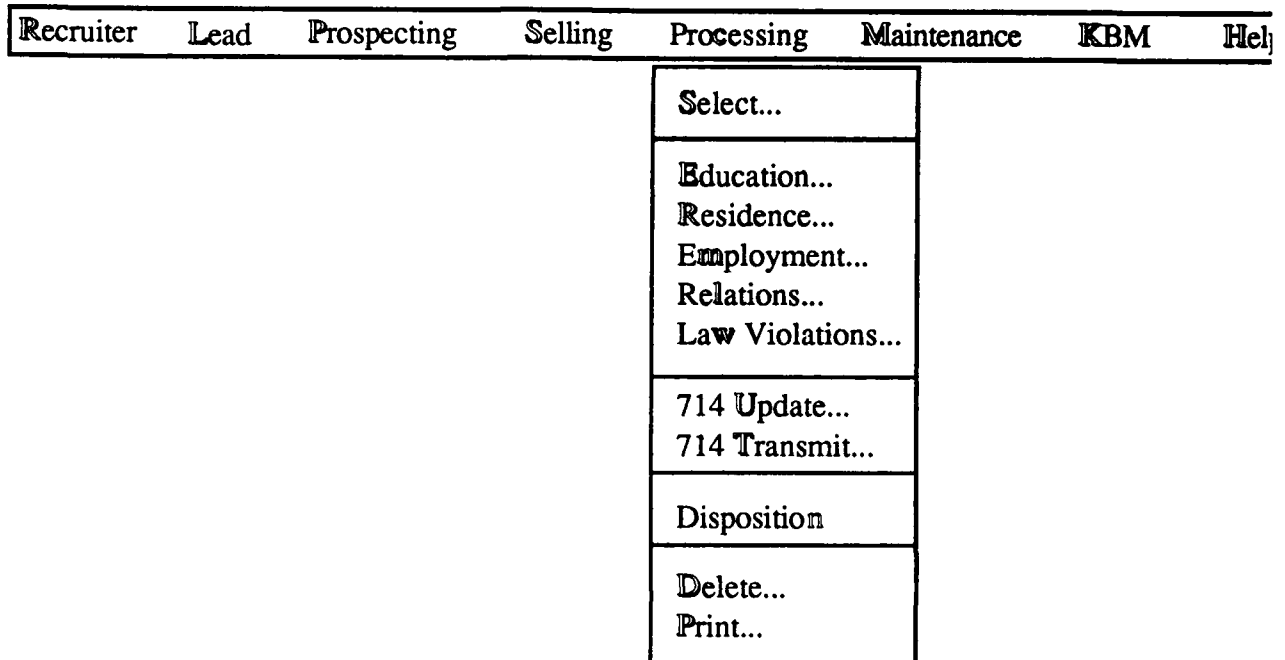


Figure 5.6. Processing Menu

The Processing Menu (Figure 5.6.) provides recruiter access to the Processing Phase options. Processing helps the recruiter fulfill all 601-210 requirements for qualifying an applicant. It also serves the following functions:

- RAPS prompts for incomplete data
- RAPS prints letters to law enforcement agencies
- RAPS prints relevant DOD forms
- RAPS transmits data records to ARADS

"Select" presents a list of the currently documented prospects and allows selection of a prospect for access. "Update" allows access to five data screens to complete the applicant data record with information regarding education, residence, employment, relations, and law violations.



"714 Update" allows Form 714A to be updated with information relevant to scheduling an applicant for his/her MEPS appearance. "714 Transmit" makes it possible to electronically transmit the applicant's data record to the MEPS.

"Disposition" keeps track of what happens to the applicant. "Delete" allows you to delete the currently selected applicant from the database. Print allows you to make a paper copy of your entire list of applicants or of a selected subset of applicants.

### 5.6.1. Select Applicant Dialog

The Applicant Select dialog of the Processing Phase presents a radio list box in which only one applicant out of many may be selected (Figure 5.6.1.). Find the name of the applicant you wish to access in the box and select it by moving to it with the Up- or Down-arrow key. The name selected is marked by an arrowhead.

Select Applicant

Select Name

Best, Lead IM  
Goodlead, John R.  
Hoyle, Arnold A.  
Nash, Jackson T.

Cancel Select

Figure 5.6.1. Select Applicant Dialog

Having selected the applicant, TAB to the "Select" radio button. Pressing the Enter key at this point will access the data relevant to the selected applicant. The hot-key Alt-S may also be used to select the designated applicant.

Pressing the hot-key Alt-C will cancel the selection. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the selection.

### 5.6.2. Update Education Data Dialog(s)

RAPS maintains all data relevant to an applicant's education. Use this dialog box (Figure 5.6.2.) for that purpose.

The dialog box is titled "Education Data". It contains the following elements:

- Three input fields labeled "From", "To", and "Grad" arranged horizontally.
- A single-line text input field labeled "School".
- A single-line text input field labeled "Location".
- A grid of six buttons arranged in two rows and three columns:
  - Top row: "Cancel", "Add", "Prev"
  - Bottom row: "Next", "Delete", "Update"

Figure 5.6.2. Update Education Dialog

Multiple schools are entered, edited, and deleted through this dialog box. Upon first entry to the Education Data dialog, data relevant to the most recent school attended are presented in the box. If there are no data, you can enter data and use the "Add" radio button to add these data to the applicant's data record.

To access data for earlier schools, TAB to the "Prev" radio button and press the Enter key. Alternatively, press the hot-key Alt-V. Each use of the "Prev" radio button will access data for earlier schools.

To access data for later schools, TAB to the "Next" radio button and press the Enter key. Alternatively, press the hot-key Alt-N. Each use of the "Next" radio button will access data for later schools.

To modify and update information relevant to any school, use the "Prev" and "Next" radio buttons to find the school, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press the Enter key. Alternatively, press the hot-key Alt-U to update the record.

To delete information relevant to any school, use the "Prev" and "Next" radio buttons to find the school, TAB to the "Delete" radio button, and press the Enter key. Alternatively, press the hot-key Alt-D to delete the current school record.

To enter data, begin with the most recent school attended and enter the year and month in which attendance was begun in the field labeled "From.". Enter the year and month in which attendance was completed in the field labeled "To.". Enter whether or not the applicant graduated in the field labeled "Grad.". Next, enter the name of the school and the school location where appropriate.

Having completed the relevant school information, TAB to the "Add" radio button and press the Enter key to "Add" this school to the applicant's record. Alternatively, press the hot-key Alt-A to add the record.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel your work.

### **5.6.3. Update Residence Data Dialog**

RAPS maintains all data relevant to an applicant's residences. Use this dialog box (Figure 5.6.3) for that purpose. This dialog allows for multiple entries, which may be necessary to document all residences for the last 5 years or since the applicant's 13<sup>th</sup> birthday, whichever is shorter.

Multiple residences are entered, edited, and deleted through the same dialog box. Upon first entry to the Residence Data dialog, data relevant to the most recent address are presented in the box. If there are no data, you can enter data and use the "Add" radio button to add these data to the applicant's data record.

To access data for earlier residences, TAB to the "Prev" radio button and press the Enter key. Alternatively, press the hot-key Alt-V. Each use of the "Prev" radio button will access data for earlier residences.

Residence Data

From yymm	To yymm
Address Street City, State ZIP 00001:001	
Cancel	Add
Next	Delete
	Update

Figure 5.6.3. Update Residence Dialog

To access data for later residences, TAB to the "Next" radio button and press the Enter key. Alternatively, press the hot-key Alt-N. Each use of the "Next" radio button will access data for later residences.

To modify and update information relevant to any residence, use the "Prev" or "Next" radio button to find the residence, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press the Enter key. Alternatively, press the hot-key Alt-U to update the record.

To delete information relevant to any residence, use the "Prev" or "Next" radio button to find the residence, TAB to the "Delete" radio button and press the Enter key. Alternatively, press the hot-key Alt-D to delete the current residence record.

To enter data, begin with the most recent residence and enter the year and month in which residence began in the field labeled "From.". Enter the year and month in which residence was terminated in the field labeled "To.". Next enter the residence address in the edit box labeled "Address".

The Address edit box is entered with the cursor in block form. While the cursor is in block form, the Tab key will allow the Address box to be skipped. Any other key will enter the Address box (the cursor will change to underscore form), and editing will be implemented in insert mode. To toggle

between insert mode and overstrike mode, use the Ins key. Two numbers, separated by a colon, appear at the bottom of the box. The first number indicates the line number being edited; the second number indicates the column. An "I" or an "O" appears when toggling the Ins key to indicate whether editing is implemented in insert mode or overstrike mode. The Address box will accommodate any number of lines.

To exit the Address box and advance to the next field, use the Tab key. You may also use Ctrl-Q to change the cursor back to block form and enable you to use the Tab key. Alternatively, use the hot-key appropriate to the radio buttons at the bottom of the dialog box.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the update.

#### 5.6.4. Update Employment Data Dialog

RAPS maintains all data relevant to an applicant's employment. Use this dialog box (Figure 5.6.4.) for that purpose. This dialog allows for multiple entries, which may be necessary to document all periods of employment and unemployment during the last 5 years.

Employment Data

From  To

Employer  Supervisor

Address   
City, State ZIP

Location

Figure 5.6.4. Update Employment Dialog

Multiple employers are entered, edited, and deleted through the same dialog box. Upon first entry to the Employment Data dialog, data relevant to the most recent employer are presented in the box. If there are no data, you can enter data and use the "Add" radio button to add these data to the applicant's data record.

To access data for earlier employers, TAB to the "Prev" radio button and press the Enter key. Alternatively, press the hot-key Alt-V. Each use of the "Prev" radio button will access data for earlier employers.

To access data for later employers, TAB to the "Next" radio button and press the Enter key. Alternatively, press the hot-key Alt-N. Each use of the "Next" radio button will access data for later employers.

To modify and update information relevant to any employer, use the "Prev" and "Next" radio buttons to find the employer, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press <Enter>. Alternatively, press the hot-key Alt-U) to update the modified record.

To delete information relevant to any employer, use the "Prev" and "Next" radio buttons to find the employer, TAB to the "Delete" radio button and press the Enter key. Alternatively, press the hot-key Alt-D to delete the current employer data.

To enter data, begin with the most recent employer and enter the year and month in which employment began in the field labeled "From:". Enter the year and month in which employment was terminated in the field labeled "To.". Next, identify the employer, then TAB to the Address field.

The Address edit box is entered with the cursor in block form. While the cursor is in block form, the Tab key will allow the Address box to be skipped. Any other key will enter the Address box (the cursor will change to underscore form), and editing will be implemented in insert mode. To toggle between insert mode and overstrike mode, use the Ins key. Two numbers, separated by a colon, appear at the bottom of the box. The first number indicates the line number being edited; the second number indicates the column. An "I" or an "O" appears when toggling the Ins key to indicate whether editing is implemented in insert mode or overstrike mode. The Address box will accommodate any number of lines.

To exit the Address box, use the Tab key to advance to the next field, or use Ctrl-Q to change the cursor back to block form and enable you to use the Tab key. Alternatively, use the hot-key appropriate to the next field(s).

Enter the location worked in the Location field. Then select the appropriate radio button at the bottom of the dialog box and press the Enter key.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel your entries.

### 5.6.5. Update Relations Data Dialog

RAPS maintains all data relevant to an applicant's relatives. Use this dialog box (Figure 5.6.5.) for that purpose. This dialog allows for multiple entries which may be necessary to document all relatives in the required categories (father, mother, spouse, and children).

Relations Data \_\_\_\_\_

Name of Relation: \_\_\_\_\_

Relation: \_\_\_\_\_ Dep y/n Birth Date: mm/dd/yy

Birth Place City, ST Country Citizenship Country Present Address City, ST Country

Cancel Add Prev Next Delete Update

Figure 5.6.5. Update Relations Dialog

Multiple relations are entered, edited, and deleted through the same dialog box. Upon first entry to the Relations Data dialog, data relevant to the closest relative are presented in the box. If there are no data, you can enter data and use the "Add" radio button to add these data to the applicant's data record.

To access data for other relatives, TAB to the "Next" or "Prev" radio buttons and press the Enter key. Alternatively, press one of the corresponding hot-keys Alt-N or Alt-V. Each use of the "Next" or "Prev" radio buttons will access data entered subsequently or earlier.

To modify and update information relevant to any relative, use the "Prev" and "Next" radio buttons to find the relative, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press the Enter key. Alternatively, press the hot-key Alt-U to update the record.

To delete information relevant to any relative, use the "Prev" and "Next" radio buttons to find the data, TAB to the "Delete" radio button, and press the Enter key. Alternatively, press the hot-key Alt-D to delete the current data.

To enter data, complete all of the relevant fields. Then TAB to the "Add" radio button and press the Enter key to "Add" these data to the applicant's record. Alternatively, press the hot-key Alt-A to add the data.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel your work.

#### **5.6.6. Update Law Violations Data Dialog**

RAPS maintains all data relevant to law violations. Use this dialog box (Figure 5.6.6.) for that purpose.

Multiple offenses are entered, edited, and deleted through this dialog box. Upon first entry to the Law Violations Data dialog, data relevant to the most recent offense are presented in the box. If there are no data, you can enter data and use the "Add" radio button to add these data to the applicant's data record.



Law Violations

Offense	Category Minor Non	Off Date mm/dd/yy
Place City, State	Age	Disposition
		Court City, State
Location of Check City, State		Date of Check mm/dd/yy
Cancel	Add	Prev
		Next
	Delete	Update

Figure 5.6.6. Update Law Violations Dialog

To access data for earlier offenses, TAB to the "Prev" radio button and press the Enter key. Alternatively, press the hot-key Alt-V. Each use of the "Prev" radio button will access data for earlier offenses.

To access data for later offenses, TAB to the "Next" radio button and press the Enter key. Alternatively, press the hot-key Alt-N. Each use of the "Next" radio button will access data for later offenses.

To modify and update information relevant to any offense, use the "Prev" or "Next" radio button to find the offense, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press the Enter key. Alternatively, press the hot-key Alt-U to update the record.

To delete information relevant to any offense, use the "Prev" or "Next" radio button to find the offense, TAB to the "Delete" radio button, and press the Enter key. Alternatively, press the hot-key Alt-D to delete the data.

To enter data, begin with the most recent offense. Enter the offense description, offense date, place of offense, age of applicant at time of offense, disposition of case, court, location of the police station where the law violation was checked, and the date of the check. Next, TAB to the "Add" radio button and press the Enter key to "Add" these data to the applicant's record. Alternatively, press the hot-key Alt-A to add the data.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel your work.

### 5.6.7. Form 714 Request for Examination Update Dialog

The Form 714 Request (Figure 5.6.7.) is used to specify information needed to request testing from a MEPS facility. As requests may be made for multiple individuals, this dialog allows the recruiter to cycle through applicants being processed and add request data as needed.

714 Request Data

Proj Date mm/dd/yy	Arrival ddhh	Comp	Proj Code Test		
Last Up mm/dd/yy	Recruiter Adams, Samuel R.		Station Code 6A32		
Tran Date mm/dd/yy	QT	Med Stat mm/dd/yy	DEP Date mm/dd/yy	Ship Date mm/dd/yy	
Cancel	Add	Prev	Next	Delete	Update

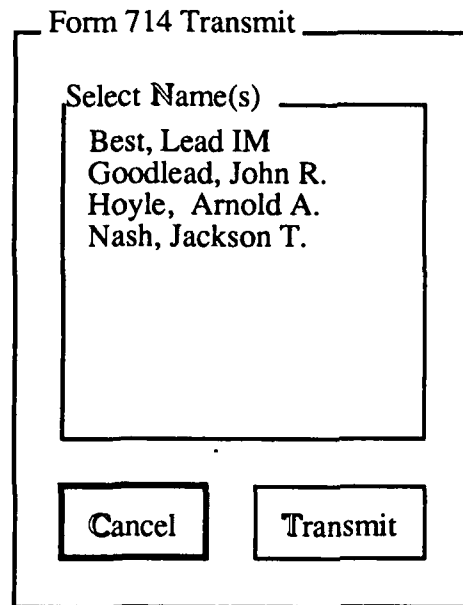
Figure 5.6.7. Form 714 Request Dialog

### 5.6.8. Transmit Form 714 Dialog

RAPS will transmit Form 714 to ARADS (Figure 5.6.8.). Select the applicants whose records are to be transmitted to ARADS by moving to them with the Up-or Down-arrow key. Select (or deselect) each name with the space bar. The selected applicants will be marked by check marks. Records transmitted are updated to indicate date and time of transmittal for later reference.

Having selected the applicants, TAB to the "Transmit" radio button and press the Enter key to transmit the selected applicant 714s from the database. You may also use the hot-key Alt-T to begin transmission.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the print.



The dialog box is titled "Form 714 Transmit". It contains a label "Select Name(s)" followed by a list of names: "Best, Lead IM", "Goodlead, John R.", "Hoyle, Arnold A.", and "Nash, Jackson T.". At the bottom of the dialog box are two buttons: "Cancel" and "Transmit".

Figure 5.6.8. Form 714 Transmit Dialog

#### 5.6.9. Disposition Dialog

RAPS maintains records of what happens to leads so that it can continually assess and update lead priority. Use the "Disposition" radio buttons to indicate lead disposition. Use the Up- or Down-arrow key to select the most appropriate disposition from the list (Figure 5.6.9.).

Disposition

<b>Disposition:</b> (*) Unwilling to commit ( ) Lead becomes prospect ( ) Found to be disqualified ( ) Moved out of zone ( ) Referred to another ( ) Unable to contact ( ) Enlisted in another service ( ) Totally without interest ( ) Plans on continuing education ( ) Attending college or other school ( ) Enlist into U.S. Army ( ) Other (See Comments)	<b>Comp:</b> <div style="border: 1px solid black; padding: 2px; text-align: center;">RA</div> <b>Comments</b> <div style="border: 1px solid black; height: 100px; width: 100%;"></div> <div style="text-align: right; margin-top: 5px;">00001:001</div>
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Cancel</div>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">Update</div>

Figure 5.6.9. Disposition Dialog

TAB to the Comments edit box to enter your comments regarding the contact. The Comments box is entered in overstrike mode. To toggle between Alt-A mode and overstrike mode, use the Ins key. To exit the edit box, use the Tab key or Shift+Tab to go to the next or previous control. Alternatively, use the hot-key Alt-U to advance to the "Update" button.

TAB to the "Update" button and push the Enter key to record your selections in the lead database. Alternatively, use the hot-key Alt-U to update the record.

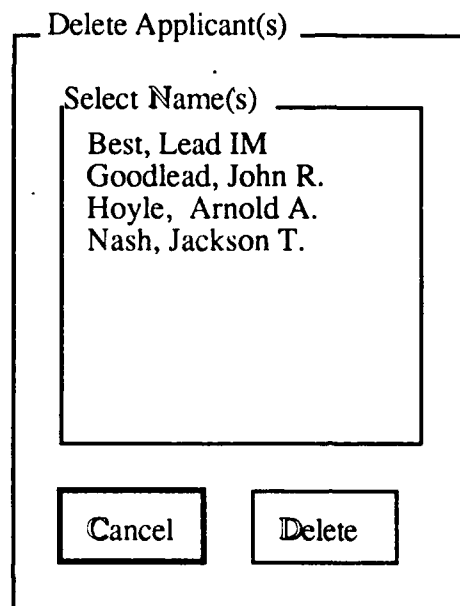
Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

#### 5.6.10. Delete Applicant(s) Dialog

The Delete Applicant(s) dialog (Figure 5.6.10.) presents a choice list box in which any number of applicants may be selected. Find the applicant(s) you wish to delete from the database by moving to them with the Up- or Down-arrow key. Select or deselect each applicant with the space bar. Selected applicants will be marked with a check mark.

Having selected the applicant's(s') name(s), TAB to the "Delete" radio button and press the Enter key to delete the selected applicants from the database. You may also use the hot-key Alt-D to delete the applicants.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the deletion.



The dialog box is titled "Delete Applicant(s)". It contains a list box labeled "Select Name(s)" with the following text: "Best, Lead IM", "Goodlead, John R.", "Hoyle, Arnold A.", and "Nash, Jackson T.". Below the list box are two buttons: "Cancel" and "Delete".

Figure 5.6.10. Delete Applicant(s) Dialog

#### 5.6.11. Print Applicant(s) Dialog

The Print Applicant(s) dialog (Figure 5.6.11.) box allows printing applicants. The Print Group box allows you to identify whether to print the entire group of applicants or only selected applicants. Use the Up- or Down-arrow key to indicate which printing option you want.

Print Applicant(s)

Print Group All	Select Name(s) Best, Lead IM Goodlead, John R. Hoyle, Arnold A. Nash, Jackson T.
Print Order by Name	
Print Form Report Form	
<div>Cancel</div> <div>Print</div>	

Figure 5.6.11. Print Applicant(s) Dialog

Next, TAB to the Print Order box to indicate whether to print applicants in alphabetic order by name, in alphabetic order by source, or in alphabetic order by priority. Use the Up- or Down-arrow key to indicate your choice. TAB to the Print Form box to indicate which forms you wish to print. Use the Up- or Down-arrow key to indicate whether you want to print a summary report of applicants, Form 200, Form 1966, Form 200B, or Form 714.

To select specific applicants to print, TAB to the Select Name(s) box and select the applicants to print by moving to them with the Up-or down- arrow key. Select (or deselect) each name with the space bar. The selected names will be marked by a check mark.

Having selected the applicants, TAB to the "Print" radio button and press the Enter key to print the selected applicants from the database. You may also use the hot-key Alt-P to print the selected applicants.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the print.

## 5.7. Enlistment Maintenance Phase

The Enlistment Maintenance Phase begins after the applicant takes his/her oath of enlistment, which is part of the Processing Phase. Enlistment maintenance is intended to reduce or minimize DEP losses during the time between enlistment and shipping by developing an ongoing interaction with the DEP/DTP recruit during that period.

Currently, DEP/DTP management is accomplished by manual maintenance of USAREC Form 200B. All interactions with the recruit during the Maintenance Phase are recorded on this form. RAPS automates these interactions.

The Maintenance Menu (figure 5.7.) gives the recruiter access to the Maintenance Phase options. These provide for selecting recruit data for update, actual information update, deleting one or more recruit records, and printing information relating to one or more recruits.

Recruiter	Lead	Prospecting	Selling	Processing	Maintenance	KBM	Help
-----------	------	-------------	---------	------------	-------------	-----	------

Select...
DEP Information...
DEP Function...
Law Violations...
DEP Loss Data...
Disposition
Delete...
Print...

Figure 5.7. Maintenance Menu

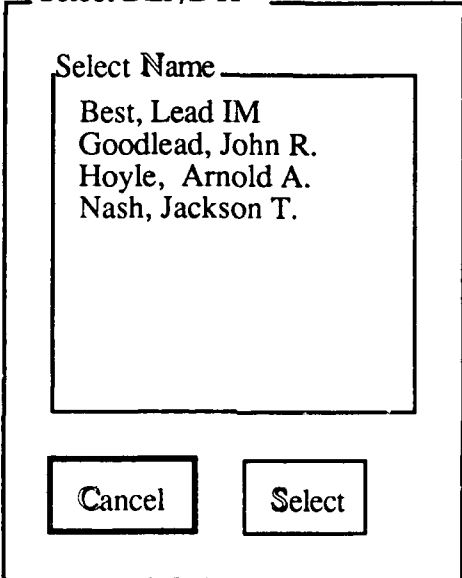
### 5.7.1. Select DEP/DTP Dialog

The Select DEP/DTP dialog (Figure 5.7.1.) presents a radio list box of those recruits currently in DEP/DTP status. Find the name of the recruit you wish

to access in the box and select it by moving to it with the Up- or Down-arrow key. The name selected is marked by an arrowhead.

Having selected the recruit, TAB to the "Select" radio button. Pressing the Enter key at this point will access the data relevant to the selected recruit. The hot-key Alt-S may also be used to select the designated recruit.

Pressing the hot-key Alt-C will cancel the selection. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the selection.

A rectangular dialog box titled "Select DEP/DTP". Inside the box, there is a label "Select Name" followed by a list of four names: "Best, Lead IM", "Goodlead, John R.", "Hoyle, Arnold A.", and "Nash, Jackson T.". Below the list, there are two buttons: "Cancel" on the left and "Select" on the right.

Select DEP/DTP

Select Name

Best, Lead IM  
Goodlead, John R.  
Hoyle, Arnold A.  
Nash, Jackson T.

Cancel Select

Figure 5.7.1. Select DEP/DTP Dialog

### 5.7.2. Update DEP/DTP Information

The DEP/DTP Information dialog (Figure 5.7.2.) provides for update of the basic data found on a USAREC Form 200B. This entry is primarily to document the initial orientation following MEPS processing and the oath of enlistment. The shipping and follow-up dates are indications of future events times. These dates are used to sort and print a schedule of events in the "Print" dialog option.

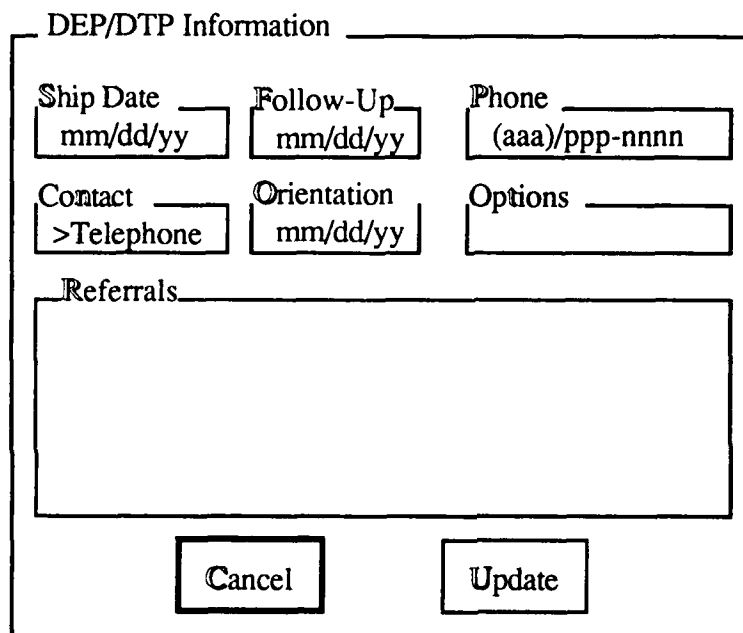
Enter a contact phone number into the Phone field. Enter the shipping date and first date of follow-up. Enter the date of DEP orientation. Enter the



enlistment options, DEP functions, and referrals generated by the new recruit.

Having updated DEP information, TAB to the "Update" radio button. Pressing the Enter key at this point will update DEP information. Alternatively, use the hot-key Alt-U to update DEP information.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the selection.



The dialog box is titled "DEP/DTP Information". It contains several input fields arranged in a grid. The first row has "Ship Date" (mm/dd/yy), "Follow-Up" (mm/dd/yy), and "Phone" ((aaa)/ppp-nnnn). The second row has "Contact" (>Telephone), "Orientation" (mm/dd/yy), and "Options". Below these is a large text area labeled "Referrals". At the bottom are two buttons: "Cancel" and "Update".

Ship Date mm/dd/yy	Follow-Up mm/dd/yy	Phone (aaa)/ppp-nnnn
Contact >Telephone	Orientation mm/dd/yy	Options
Referrals		
Cancel		Update

Figure 5.7.2. DEP/DTP Information Dialog

### 5.7.3. Update DEP/DTP Function Data

During the DEP/DTP phase, there may be many interactions between the recruiter and the recruit. The DEP/DTP Function dialog (Figure 5.7.3) allows recording of these events. As many events as necessary may be entered. Also, previously entered DEP/DTP events may be reviewed and edited if necessary.

DEP/DTP Function \_\_\_\_\_

Location \_\_\_\_\_ Date  Ht  Wt

Remarks

00001:001

Figure 5.7.3. DEP/DTP Function Dialog

#### 5.7.4. Update DEP/DTP Law Violations

The DEP/DTP Law Violations dialog (Figure 5.7.4.) provides for entry of any law violations that occur during the DEP/DTP Maintenance Phase. Multiple offenses are entered, edited, and deleted through this dialog box. Upon first entry to the Law Violations Data dialog, data relevant to the most recent offense are presented in the box. If there are no data, you can enter data and use the "Add" radio button to add these data to the applicant's data record.

Law Violations

Offense	Category Minor Non	Off Date mm/dd/yy
Place City, State	Age	Disposition
		Court City, State
Location of Check City, State		Date of Check mm/dd/yy
Cancel	Add	Prev
		Next
		Delete
		Update

Figure 5.7.4. Update Law Violations Dialog

To access data for earlier offenses, TAB to the "Prev" radio button and press the Enter key. Alternatively, press the hot-key Alt-V. Each use of the "Prev" radio button will access data for earlier offenses.

To access data for later offenses, TAB to the "Next" radio button and press the Enter key. Alternatively, press the hot-key Alt-N. Each use of the "Next" radio button will access data for later offenses.

To modify and update information relevant to any offense, use the "Prev" or "Next" radio button to find the offense, TAB to the field in error, correct the field, TAB to the "Update" radio button, and press the Enter key. Alternatively, press the hot-key Alt-U to update the record.

To delete information relevant to any offense, use the "Prev" or "Next" radio buttons to find the offense, TAB to the "Delete" radio button and press the Enter key. Alternatively, press the hot-key Alt-D to delete the data.

To enter data, begin with the most recent offense. Enter the offense description, offense date, place of offense, age of applicant at time of offense, disposition of case, court, location of the police station where the law violation was checked, and the date of the check. Next, TAB to the "Add" radio button and press the Enter key to "Add" these data to the applicant's record. Alternatively, press the hot-key Alt-A to add the data.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel your work.

#### 5.7.5. Update DEP/DTP Loss Data

In instances where DEP/DTP loss occurs the Update DEP/DTP Loss dialog (Figure 5.7.5.) allows documenting the circumstances surrounding the loss. Completion of this dialog will effectively remove the recruit from the DEP/DTP Maintenance Phase and place the recruit in an inactive status.

The diagram shows a rectangular dialog box with a title bar at the top labeled "DEP/DTP Loss". Below the title bar is a text input field labeled "Reason". Underneath the "Reason" field are two date input fields: "CC Interview mm/dd/yy" on the left and "Loss Date mm/dd/yy" on the right. At the bottom of the dialog box are two buttons: "Cancel" on the left and "Update" on the right.

Figure 5.7.5. DEP/DTP Loss Dialog

Use the DEP/DTP Loss Dialog box to indicate the reason for the loss. Enter the date of the termination interview, the recruiter's name, and the effective loss date.

TAB to the "Update" radio button and press the Enter key to record the DEP loss. Alternatively, use the hot-key Alt-U to record the loss.

Pressing the Enter key when the "Cancel" radio button is selected will cancel the action. Likewise, the hot-key Alt-C will cancel the action of recording the DEP loss

#### 5.7.6. Disposition Dialog

RAPS maintains records of what happens to leads so that it can continually assess and update lead priority. Use the "Disposition" radio buttons to

indicate lead disposition. Use the Up- or Down-arrow key to select the most appropriate disposition from the list (Figure 5.7.6.).

Disposition

<p>Disposition:</p> <ul style="list-style-type: none"><li>(*) Unwilling to commit</li><li>( ) Lead becomes prospect</li><li>( ) Found to be disqualified</li><li>( ) Moved out of zone</li><li>( ) Referred to another</li><li>( ) Unable to contact</li><li>( ) Enlisted in another service</li><li>( ) Totally without interest</li><li>( ) Plans on continuing education</li><li>( ) Attending college or other school</li><li>( ) Enlist into U.S. Army</li><li>( ) Other (See Comments)</li></ul>	<p>Comp: <input type="text" value="RA"/></p> <p>Comments <input type="text"/></p> <p>00001:001</p> <p><input type="button" value="Cancel"/> <input type="button" value="Update"/></p>
--	---

Figure 5.7.6. Disposition Dialog

TAB to the Comments edit box to enter your comments regarding the contact. The Comments box is entered in overstrike mode. To toggle between insert mode and overstrike mode, use the Ins key. To exit the edit box, use Tab or Shift+Tab to go to the next or previous control. Alternatively, use the hot-key Alt-U to advance to the "Update" button.

TAB to the "Update" button and push the Enter key to record your selections in the lead database. Alternatively, use the hot-key Alt-U to update the record.

Pressing the hot-key Alt-C will cancel the update. Pressing the Enter key while the "Cancel" radio button is selected will also cancel the update.

### 5.7.7. Delete DEP/DTP(s)

The Delete DEP/DTP dialog (Figure 5.7.7.) presents a choice list box in which one or more recruits may be identified for deletion. Find the

recruit(s) you wish to delete from the database by moving to them with the Up- or Down-arrow key. Use the space bar to mark the selected recruits with a check mark. Toggling the space bar marks and unmarks the selected recruit.

Having selected the recruit names, TAB to the "Delete" radio button and press the Enter key to delete the selected recruits from the database. You may also use the hot-key Alt-D to delete the recruits.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the deletion.

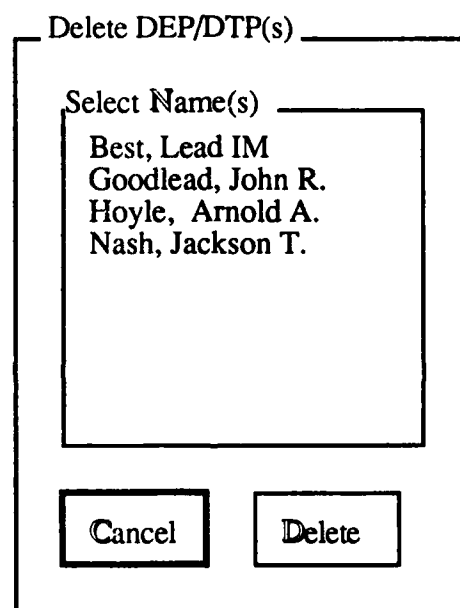


Figure 5.7.7. Delete DEP/DTP(s) Dialog

### 5.7.8. Print DEP/DTP(s) Dialog

The Print DEP/DTP dialog (Figure 5.7.8.) allows printing recruit data. The Print Group box allows you to identify whether to print data on all recruits or selected recruits. Use the Up- or Down-arrow key to indicate which printing option you want.

Print DEP/DTP(s)

Print Group All	Select Name(s) Best, Lead IM Goodlead, John R. Hoyle, Arnold A. Nash, Jackson T.
Print Order by Name	
Print Form Report Form	
<div>Cancel</div> <div>Print</div>	

Figure 5.7.8. Print DEP/DTP(s) Dialog

Next, TAB to the Print Order box to indicate whether to print recruits in alphabetic order by name, in alphabetic order by source, or in alphabetic order by priority. Use the Up- or Down-arrow key to indicate your choice.

TAB to the Print Form box to indicate which forms you wish to print. Use the Up- or Down-arrow key to indicate whether you want to print a summary report of applicants, Form 200, Form 1966, Form 200B, or Form 714.

To select specific recruits to print, TAB to the Select Name(s) box and select the recruits to print by moving to them with the Up-or down- arrow key. Select (or deselect) each name with the space bar. The selected names will be marked by a check mark.

Having selected the recruits, TAB to the "Print" radio button and press the Enter key to print the selected recruits from the database. You may also use the hot-key Alt-P to print the selected recruits.

Pressing the hot-key Alt-C or pressing the Enter key while the "Cancel" radio button is selected will cancel the print.



## 6. RAPS Knowledge Base

The RAPS knowledge base is used to verify and validate recruiter entries. The knowledge base is also used to qualify various applicant parameters, such as age, citizenship, education, etc.

The knowledge base contains rules derived from AR 601-210 as well as other supporting USAREC regulations, such as the Recruiter Production Management System (USAREC Reg. 350-6).

### 6.1 Overview

As a recruiter enters data in the various dialog boxes during the different phases of recruiter access, lead generation, prospecting, selling, processing and maintenance, certain entries must be verified, validated, or qualified. These data are checked by the knowledge base.

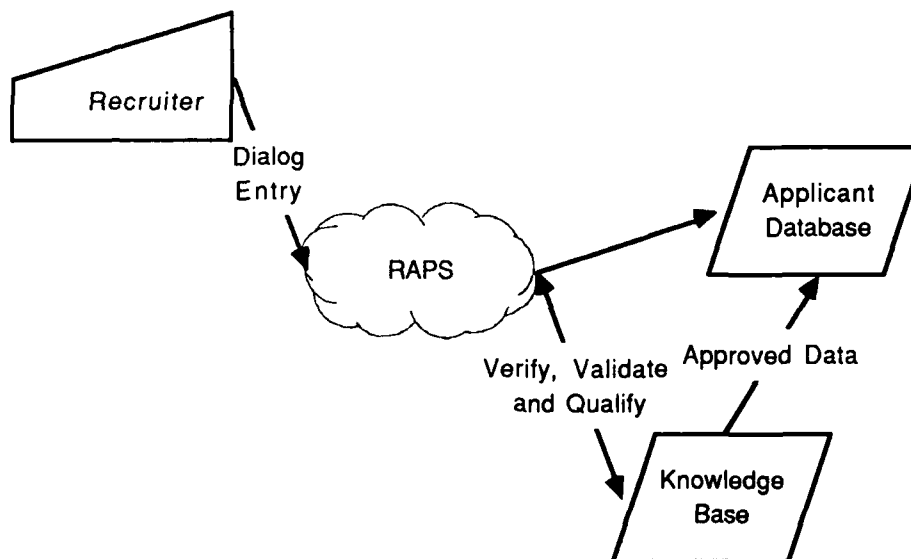


Figure 6.1. RAPS Knowledge Base Context

The RAPS knowledge base contains seven types of information. First, the knowledge base contains the valid entries possible for the dialog box data items, that, is those items that need to be validated on entry (validation rules). Second, the knowledge base contains verification rules; these define the ways in which certain data items may be verified. Third, the knowledge base contains qualification rules, based on regulations; these define how applicants

qualify for military service. Fourth are waiver rules that define how and when waivers can be used to qualify applicants. Fifth are form rules, which relate applicant data to relevant DA and DOD forms. Sixth, the knowledge base contains data rules; these define where in the database to obtain information relevant to the rules. Seventh, the knowledge base contains templates which define the format in which data must be entered.

For example, there are many rules relevant to applicant age. Validation rules are used to prevent errors from being entered by the recruiter. These rules may include the fact that age is a numeric field that cannot have values greater or less than certain specified values. Verification rules define how the recruiter can verify the applicant's age (e.g., drivers license, birth certificate, etc.). Qualification rules define whether or not the applicant is age-qualified for enlistment (e.g., age > 17 and < 35). A waiver rule is invoked if the applicant is 17 (i.e. a parental consent form is required) or if the applicant is 35 or older (prior service). Form rules can be used to print a parental consent form or to complete prior service information. Data rules define how to obtain the applicant's age from the applicant's data record (subtract applicant's date of birth from the present date). And finally, Format rules specify how dates, times, test scores, etc. must be entered. For example, a format rule can define a like the following: "YYMMDD."

## 6.2 Knowledge Base Maintenance

The regulations that form the basis for the RAPS knowledge base are continuously evolving. Situations change that cause existing regulations to become dated. Also, existing regulations continue to be interpreted and clarified. For the RAPS knowledge base to maintain validity, it must be updated as regulations change and interpretations evolve.

A program facility, KBM, is provided to allow ongoing development of the RAPS knowledge base. KBM provides menu and dialog box options for continued maintenance of the the RAPS knowledge base.

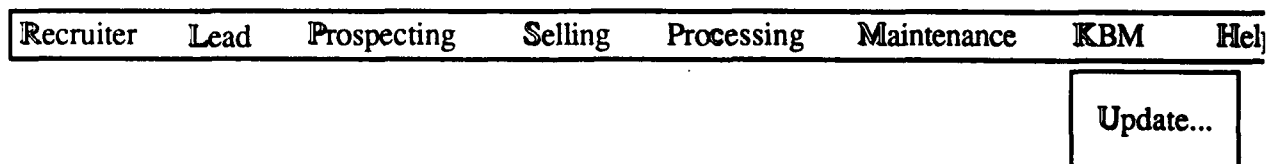


Figure 6.2(a). RAPS Knowledge Base Maintenance Menu



The first choice in the Rule Select field is always labeled "new\_rule" [Figure 6.2(c)]. It is used to enter new rules into the knowledge base. Selecting "new\_rule" transfers control to the Rule Box where the new rule may be entered. By Tabbing to the Information box, rule information and rule clarifications may be entered.

It is important to note that by adding new rules, the user can add rule types beyond the seven already defined. Each new rule requires a type designation and the user can create new rule types as needed.

Selecting any item from Rule Select other than "new\_rule" will display the given rule in the field labeled "Rule." Here the rule may be modified. Rule information and rule clarifications may be added in the field labeled "Information."

Update Rule

<p>Rule Type</p> <p>&gt;Qualify</p>	<p>Rule</p> <pre>rule(q5,qualify) ?? age_ok if [(dbt(pLVAge,A) , ! , A&gt;=18 rule(q6,qualify) ?? age_ok if [(dbt(pLVAge,A) , ! , A =:= 1 rule(q7,qualify) ?? age_ok if [(dbt(pLVAge,A) , ! , A&lt;55),</pre>
<p>Rule Select</p> <ul style="list-style-type: none"> <li>new_rule</li> <li>&gt;age_ok</li> <li>qual601_210</li> <li>qual_age</li> <li>qual_basic</li> <li>qual_citizenship</li> <li>qual_dependents</li> <li>qual_mental</li> <li>qual_moral</li> <li>qual_physical</li> </ul>	<p>Information</p> <p>00001:001</p>

00001:001

Figure 6.2(c). Update Rule Dialog/Rule Select

Having completed rule and information editing, the rule may be added by using the "Add" radio button at the bottom of the dialog box, deleted with the "Delete" radio button, updated with the "Update" button, printed with the "Print" radio button, or cancelled with the "Cancel" button. One of these choices (Add, Delete, Update, Print, or Cancel!) can be selected by Tabbing to it, or by using the respective hot-key.

### 6.3. Knowledge Base Rule Format

There are four essential elements in a RAPS rule. First, all rules must be unique so they may be processed by the Inference Engine. To assure uniqueness, an identification name or number is an art of the rule format. Second, rules may be assigned to be of a particular type for the convenience of the user in finding and grouping similar rules. Currently RAPS identifies seven rule types (Data Rules, Form Rules, Qualification Rules, Format Rules, Validation Rules, Verification Rules, and Waiver Rules). The existing rule types are not meant to be limiting and may be freely added to or deleted according to user requirements. Third, rules must identify a goal which is to be proved or concluded. Fourth, the rule contains a set of arguments which if true, imply the premise. These arguments are usually referred to as the rule premise.

Rules have the following format:

rule(ID,RuleType) := Goal if [ListOfPremises].

Rules are identified as the PROLOG predicate, "rule()". The first argument to rule is the unique rule ID, and the second argument is RuleType. RuleType can refer to any of the predefined rule types, or the user may enter new ones to reflect the knowledge base requirements. RuleType is followed by a colon and equal sign "==" and then by a Goal to be proven by the Inference Engine. The goal is followed by the word "if" which in turn is followed by a list of premises enclosed in left and right brackets. The list of premises can be of any length; each of the premises is separated by a comma, implying a logical OR. The list of premises refers to all of the IF qualifiers which make the goal true.

Consider, for example the following three rules, all of which have the goal of determining if the applicant's age is appropriate.

```
rule(q5,qualify) := age_ok if [dbv(Age,A), A>=18,, A=<34)].  
rule(q6,qualify) := age_ok if [(dbv(Age,A), A:=17),  
                                guardian_consent)  
rule(q7,qualify) := age_ok if [prior_service, (dbv(Age,A), A=<55),  
                                retirement_qual_by_age_60].
```

The first rule defines age to be appropriate under the single condition (IF) there is a database variable Age which is set equal to A "dbv(Age,A)" and that A is greater than or equal to 18 and that A is equal to or less than 34.

The second rule defines age to be appropriate under two conditions (IFs); (1) that there is a database variable Age which is set equal to A "dbv(Age,A)" such that A equates to 17, and (2) that there is guardian\_consent.

The third rule defines age to be appropriate under three conditions. Age is appropriate if (1) the applicant has had prior service, and (2) if there is a database variable Age which is set equal to A "dbv(Age,A)" and A is less than or equal to 55, and (3) if the applicant qualifies for retirement by age 60.

The format above describes the rule format as it is maintained in the knowledge base, or the runtime file named RAPSKBR.ARI. To maintain a distinction between rules already in the system and rules being edited or added through the Update Rule Dialog, the format accessible through the Update Rule Dialog differs slightly. In the Update Rule Dialog, the Rule Type is followed by two consecutive question marks as opposed to the colon and equal sign.

#### **6.4. Rules and Inference Engine Control**

The knowledge base rules can be fired from almost any point in a RAPS session. From which point rules are fired is determined by generic goals. The knowledge base rules can be accessed whenever a database record is read using the goal "expand":

```
rule(ID,RuleType) := expand if [ListOfPremises].
```

Knowledge base rules can also be accessed whenever a database record is written with the goal "collect":

`rule(ID,RuleType) := collect if [ListOfPremises].`

The Inference Engine can be accessed whenever a record is selected by using "select" as the goal:

`rule(ID,RuleType) := select if [ListOfPremises].`

To invoke the Inference Engine and access rules upon exit from a dialog box, rules with the dialog box name as the goal are used:

`rule(ID,RuleType) := dialogBoxName if [ListOfPremises].`

And, the Inference Engine can be invoked whenever a variable is accessed by RAPS by using the variable name as the goal:

`rule(ID,RuleType) := variableName if [ListOfPremises].`

Once the Inference Engine is invoked, any other set of rules can be accessed simply by having subsequent rule goals appear as premises in calling rules.

When `expert()` is invoked, it is passed the entire list of premises. Immediately, `expert()` begins to prove each element in the list. Each element in the list is sequentially accepted as a goal, and for each of these new goals, `expert()` searches through the rule base to see if there is a rule that proves the current goal. If there is, `expert()` invokes that rule and concludes the goal.

If no rule exists which allows concluding the current goal, then `expert()` looks to see if the premise has been replied to earlier. That is, `expert()` looks to see if the Inference Engine has already asked the user for a value, or for a "yes" or "no" conclusion. If there has been a reply, `expert()` uses the previous reply and continues to backtrack through the rule set to conclude the current goal.

If no rule exists which allows concluding the current goal, and if there have been no user replies to `expert()` questions, then `expert()` questions the user regarding the premise. Once answered, the reply is recorded, and `expert()` continues working through the rule base until the goal is concluded or disproven.

There is one noteworthy exception to this process. It is possible that the ListOfPremises does not contain a premise, but rather contains an atom. In this case expert() treats the atom like PROLOG code; calls the code, and behaves appropriately to what PROLOG returns.

### **6.5. The RAPS Format Rule**

RAPS implements two methods for controlling the data entered into a field to trap errors as they occur and eliminate the errors before they are stored in the RAPS database. The first method involves writing rules with variable names as goals. The second method involves invocation of the special format rule.

The format rule has the following form:

```
rule(ID,format) := Goal if
                        [format("Template",VariableName)].
```

The Goal in the format rule controls when the rule is fired and will typically be a dialog box name. The Template is formed from a string of characters that define the structure of the variable. For example, a template for a date field might be constructed as follows "99/99/99". Similarly a template for a two digit numeric value might be constructed as "99". Figure 6.5 summarizes the possible template symbols and their values.



Symbol	Description
A	Accepts only alphabetic characters A-Z and a-z and no numbers, spaces, or punctuation
9	Accepts real numbers only, including plus and minus signs
#	Accepts numeric digits 0-9, spaces, periods (.), and plus (+) and minus (-) signs
X	Any characters are accepted
Other	Any other characters are <i>literal</i> characters and are actually stored in the database

Figure 6.5. Characters Used in Field Templates

In addition to examining the entries in the RAPS database for format inconsistencies, the format rule can examine current field values in the dialog box just exited. Any current field value is checked with a template by replacing the variable name with "curfval".

```
rule(ID,format) ?? Goal if [format("Template",curfval)].
```

## 6.6. RAPS Inference Engine Interface

User interaction with the RAPS knowledge base occurs through the RAPS Inference Engine. Presently, the Inference Engine is invoked at the close of each dialog if an "Add" or an "Update" has occurred. In addition, the Inference Engine is invoked when a data record is read or written. Upon invocation, the Inference Engine looks to see if there are goals associated with the invocation. If such goals exist, the Inference Engine is started on the sequence of proving the goal(s).

Upon a meaningful exit from a dialog box (either "Add" or "Update"), the Inference Engine looks for goals associated with each variable which was set

or modified. These goals are processed first. Next, the Inference Engine looks for goals associated with the dialog box name. These goals are processed next.

When a data record is read, it is parsed with the `expand()` predicate. After expansion, control is passed to the Inference Engine to see if there are any goals labelled "expand". If such goals exist, the Inference Engine is started on the sequence of proving the goal(s).

When a data record is written, the variables are processed with the `collect()` predicate. Subsequently, control is passed to the Inference Engine which looks for all goals labelled "collect". If such goals exist, the Inference Engine begins to work through the sequence of rules with the "collect" goal to prove them.

expert

Is it true that [goal]  
yes, no, why?

Figure 6.6. Expert Inference Engine Interface Window

Once the Inference Engine is invoked, all program control passes to the Inference Engine. If the current goal is proved, and there are no additional goals, the Inference Engine passes control back to RAPS and the entire operation remains transparent to the user. If the current goal needs information that is unavailable from the database, the Inference Engine halts and queries the user. The Inference Engine may ask for a specific value for a variable (in which case that value is subsequently entered into the database), or for an answer ("yes", "no", or "why") [Figure 6.6].

The Inference Engine continues firing relevant rules and querying the user when necessary until it succeeds or fails. If the Inference Engine succeeds in

proving the goal, then control is passed back to RAPS, and the normal RAPS session continues. If the Inference Engine fails, it notifies the user and requires a response (either an acknowledgement, or the user's query as to how and why the goal was not satisfied).

If the user asks "why" at any point, the Inference Engine will work through the logic sequence to show which rules fired, and how each rule was satisfied. At each rule, the user must press the Return key to continue.

RAPS rules may be quite complex. The rule structure allows for an "IF" premise (List\_of\_Arguments) and a "THEN" conclusion (RuleName). The single conclusion, however, may depend on an entire list of premises (List\_of\_Arguments). Each element in the list is either an atom, or something else. If it is an atom, the Inference Engine looks for other rules which conclude the atom, and sequentially fires these rules in an attempt to prove the goal (conclusion). If the element is an atom which is not provable by existing rules, the Inference Engine asks the user questions to prove or disprove the goal. If the element is not an atom, then the Inference Engine assumes that it is evaluable, and retrieves the necessary information from the database to prove or disprove the goal.

## **7. RAPS Software Organization**

RAPS was developed with Arity PROLOG and is being delivered with the full version of Arity. Maintenance, as well as changes and additions, can be completed with Arity.

ICS has prepared two different versions of RAPS: a development (interpreted) version and a distribution version. The development version contains all of the source code in 12 files, and can be modified and expanded as required. The distribution version presents RAPS in a form suitable for distribution; it is prepared by compiling seven files into an executable file, RAPS.EXE. In addition, the distribution version contains RAPS.IDB, a file that contains all of the uncompiled code (run time code) as well as the RAPS databases. The RAPS files are presented below.

### **7.1. RAPS Compiled Files**

#### **7.1.1. RAPS.ARI - Raps Main Program**

The RAPS main program contains the entry point for RAPS. An initial check is made to see if the dialogs have been saved in the database (RAPS.IDB). If the dialogs are not found, they are loaded and saved. The main program then defines the top and bottom windows as well as the Main Window, which is resized to fit between the top and bottom windows. Finally, the program enters a loop that examines keystrokes and executes the appropriate RAPS functions until RAPS is terminated by the user.

RAPS contains `do_mnu` commands for handling menu selections: `do_mnu` invokes the user selected menu option. If the selection is an exit from the menu, then the selection is a cancel that is deleted, and a return is executed. If a menu option is selected, it is translated to the dialog name to be executed and the corresponding dialog is executed. After each dialog, a check is made for the quit selection. If quit has not been selected, then the user is returned to the same menu in pull-down mode for the next selection.

### **7.1.2. RAPSDCV.ARI - Dialog Control Variables**

RAPSDCV contains a table that indicates (for each accessible dialog control) the dialog control number, the control type, and the variable name. In addition, RAPSDCV contains a predicate that translates from the dialog name and control number to the variable name and variable value. Finally, RAPSDCV contains the default values for dialog variables.

### **7.1.3. RAPSDHF.ARI - Dialog Handler Functions**

RAPSDHF processes the relevant messages sent to and from dialog boxes. Message processing allows RAPS to provide context-sensitive help and keep track of which options were selected in radio and list boxes.

### **7.1.4. RAPSDHS.ARI - Dialog Handler Support Functions**

RAPSDHS provides support for RAPSDHF. The functions in RAPSDHS allow for btree storage of records, data collection for data records from dialog boxes, dialog box construction from data records, setting menu states, recording keystrokes, saving data records, and editing variables.

### **7.1.5. RAPSKBI.ARI - Knowledge Base Inference Engine**

RAPSKBI contains the predicate definitions that define the expert system Inference Engine and the interface between the Inference Engine and the RAPS databases.

### **7.1.6. RAPSRSD.ARI - Record Sequence Definitions**

RAPSRSD contains the definitions for the recruiter and applicant databases as well as the knowledge base. The record definitions indicate which variables are

contained in the database records. These definitions can be used to create value facts when records are made current or updated information is added to the records.

#### **7.1.7. RAPSUTL.ARI - Utility Predicates File**

RAPSUTL contains the predicate definitions that produce the RAPS environment. These predicates display messages, erase and record values, return menu scan codes, position windows, etc.

### **7.2. RAPS Run Time Files**

#### **7.2.1. RAPSDDC.ARI - Default Dialog Choices (run time)**

RAPSDDC contains the default choices for radio lists and radio buttons used in the RAPS dialog boxes.

#### **7.2.2. RAPSDLG.ARI - Dialog Definitions (run time)**

RAPSDLG contains the dialogs displayed as a result of RAPS menu selection.

#### **7.2.3. RAPSDLG.HLP - Dialog Help File (run time)**

RAPSDLG HLP provides for the context-sensitive help dialog box. Currently, the help is provided at the dialog box level, but this can be easily extended to the level of each control within each dialog box.

#### **7.2.4. RAPSKBR.ARI - Knowledge Base Rules (run time)**

RAPSKBR provides the knowledge base rules necessary for RAPS. All rules regarding applicant qualification, data verification, data validation, waivers, and forms are contained in RAPSKBR.

#### **7.2.5. RAPSMNU.ARI - Menu Definition (run time)**

RAPSMNU provides the menu definitions (both Menu Bar and Pull-Down) in the RAPS environment. All menu messages associated with the RAPS menus are unique to allow identification of the Pull-Down Menu from which the item was selected.

#### **7.2.6. RAPSMNU.HLP - Menu Help File (run time)**

RAPSMNU.HLP provides context-sensitive menu help.

### **7.3. Running RAPS from the Arity Interpreter**

To run RAPS interactively, invoke the Arity Interpreter:

```
C:\ARITY >api
```

After the Arity Interpreter is loaded, the user is presented with the question mark and hyphen prompt:

```
?-
```

Consult the primary RAPS file by typing:

```
?-consult(raps).
```

in response to the Interpreter prompt. Next, type:

```
?-interp.
```

to consult the RAPS support files. Finally, type:

?-main.

to begin execution at the main entry point for RAPS.

#### **7.4. Running Compiled RAPS from DOS**

The distribution or compiled version of RAPS contains only two files, RAPS.EXE and RAPS.IDB. To invoke RAPS, log onto the disk that contains both RAPS files and type:

A: >RAPS

#### **7.5. RAPS Database Structure**

The RAPS databases consist of random access, variable length records accessed by btree keys. The variable length record feature allows for minimum length records. Btree random access allows quick retrieval by name and maintains records in alphabetic order in the selection lists.

Each record in the database consists of a list of elements, each element of which contains two components: the first component is the variable name; the second component is the data. The data may be a literal string or another list (for example, the data from an edit box may contain multi-line data like an address, and such data would be represented as a list of strings, one for each line of data).

The data records for recruiters and applicants are the same in concept. Both sets of data employ predicates that instantiate variables from a data record, and that assemble data records from variables.

Records are stored under five different keys that correspond to the Processing Phase as well as record type. Recruiter data records are stored under the key uName (User Name). Lead data are stored under the key lName (Lead Name). Data for applicants in the Selling Phase are stored under the key sName (Selling Name), and data for applicants being processed are maintained under the key pName (Processing Name). Data for DEPs/DTPs are maintained under the key mName (Maintenance Name). As



leads become prospects, applicants, and recruits, their records are moved from one key to the next so that they may be found in the menu associated with the proper Processing Phase.

## 7.6. RAPS Program Architecture

RAPS is hierarchically programmed. Arity PROLOG defines the syntax and fundamental predicates. All remaining RAPS predicates are defined hierarchically, as shown in Figure 7.6. That is, all of the non-standard predicates used in the file RAPS.ARI are defined in the files RAPSDHF.ARI and RAPSUTL.ARI. All of the predicates used by RAPSDHF.ARI are defined in the file RAPS.ARI. And, as shown, the predicates used by RAPSDHF.ARI are defined in the files RAPSDHS.ARI, RAPSKBI.ARI, RAPSDCV.ARI, RAPSRS.D.ARI, and RAPSUTL.ARI.

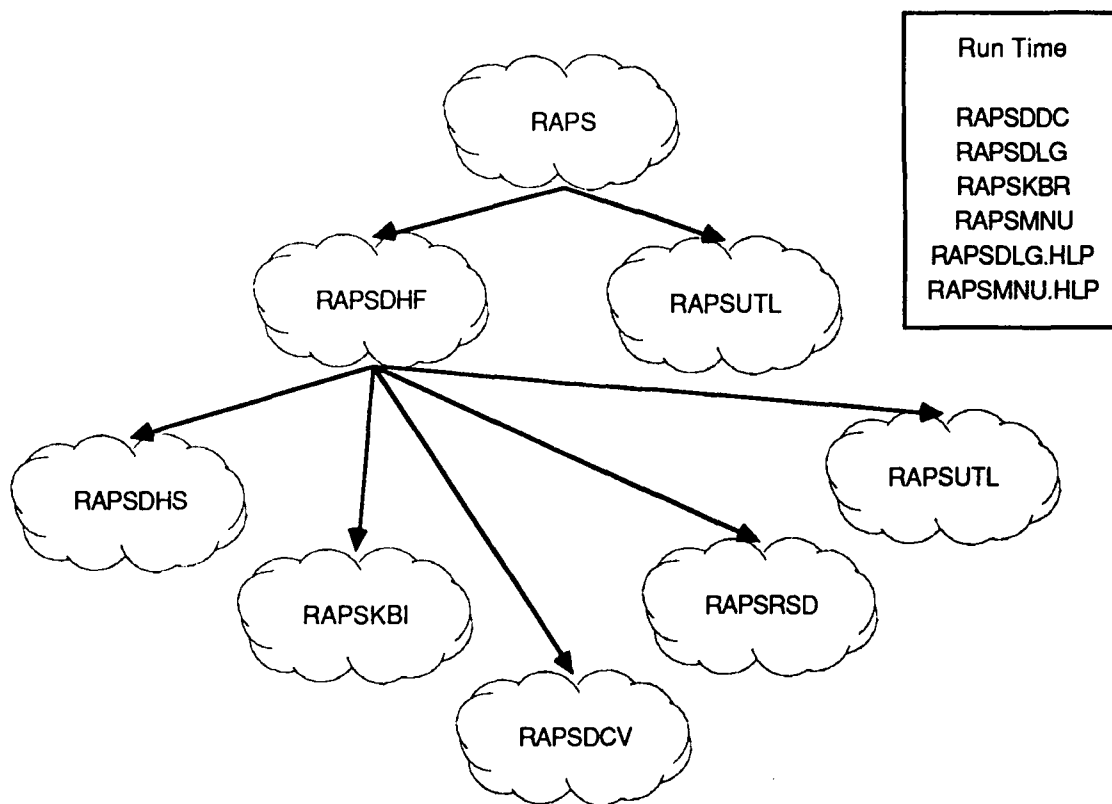


Figure 7.6. RAPS Program Architecture

### 7.6.1. RAPS Global Variables and Predicates

Global variables are accessible from any RAPS procedure or predicate. The global variables include the following:

\$dialog_help	- key under which RAPSDLG.HLP file is loaded
\$key	- key representing last selected menu option
\$larg_idx	- key under which list of lists index is stored
\$menu	- last selected menu message
\$menu_help	- key under which RAPSMNU.HLP file is loaded
\$prtMsg	- messages to be displayed in Main Window
\$quit	- flag indicating whether quit has been selected
\$select_key	- key under which data select menu key is stored
\$select_mnu	- key under which selected menu name is stored
\$select_name	- key under which data selected name is stored

Menu and dialog names are also globally defined. Below are the menu and dialog names defined in RAPS:

helpDialog	- Dialog Help dialog
helpMenu	- Menu Help dialog
kbnUpdate	- Update Rule dialog
leadAdd	- Add Lead dialog
leadDelete	- Delete Lead(s) dialog
leadImport	- Import Leads dialog
leadPrint	- Print Lead(s) dialog
leadSelect	- Select Lead dialog
leadUpdate	- Update Lead dialog
mntDelete	- Delete DEP/DTP(s) dialog
mntFollowUp	- DEP/DTP Follow-Up dialog
mntInfo	- DEP/DTP Information dialog
mntLoss	- DEP/DTP Loss dialog
mntPrint	- Print DEP/DTP(s) dialog
mntSelect	- Select DEP/DTP dialog
proc714Data	- 714 Request Data dialog
proc714Tran	- Form 714 Transmit dialog
procDelete	- Delete Applicant(s) dialog
procEduca	- Education Data dialog
procEmploy	- Employment Data dialog
procPrint	- Print Applicant(s) dialog
procRelat	- Relations Data dialog
procResiden	- Residence Data dialog

procSelect	- Select Applicant dialog
prosContacts	- Lead Contacts dialog
prosDisposition	- Lead Disposition dialog
prosPriority	- Lead Priority dialog
raps_menu	- RAPS menu definition
recChangeLogin	- Change Recruiter Login dialog
recDelete	- Delete Recruiters dialog
recLogin	- Recruiter Login dialog
recQuit	- Quit RAPS dialog
recRegister	- Register Recruiter dialog
sellCitizen	- Citizenship Data dialog
sellDelete	- Delete Prospect(s) dialog
sellIntvRem	- Interview Remarks dialog
sellLawViol	- Law Violations dialog
sellPersData	- Personal Data dialog
sellPrequal	- Prequalification Data dialog
sellPrint	- Print Prospect(s) dialog
sellPriorSvc	- Prior Service Data dialog
sellProspect	- Prospect Data dialog
sellSelect	- Select Prospect dialog
sellTestData	- Test Data dialog
sellTestDataP	- Physical Test Data dialog

Dialog variable names are the names associated with list box and edit box definitions. The variables used in the dialog definitions within RAPSDLG are as follows:

\$dialog_help	- Dialog Help
\$menu_help	- Menu Help
aEmpAddress	- Applicant Employer Address
aResAddress	- Applicant Residence Address
kbmRuleInfo	- Knowledge Base Rule Information
kbmRuleInst	- Knowledge Base Rule Instance
kbmRuleName	- Knowledge Base Rule Name
kbmRuleType	- Type of Knowledge Base Rule
lAddress	- Lead Address
lContactCode	- Lead Contact Code
lContactComment	- Lead Contact Comments
lDispCmnts	- Lead Disposition Comments
lName	- Lead Name
lSourceText	- Lead Source
pAddress	- Prospect Address
pBirthPlace	- Prospect Birth Place

pCitiz	- Prospect Citizenship
pEdCode	- Prospect Education Code
pHealth	- Prospect Health
pIntvRem	- Prospect Interview Remarks
pLastSchool	- Prospect Last School
pMarStat	- Prospect Marital Status
prtForm	- Print Form
prtGroup	- Print Group
prtOrder	- Print Order
pSex	- Prospect Sex
pTType	- Prospect Test Type
rFUpRemarks	- Recruit Follow-Up Remarks
uName	- Recruiter Name (user name)

Predicates described below are those that appear in RAPS code that are not Arity-supplied standard predicates. The file in which the predicate is described is presented to the right of each predicate. The description of each predicate includes a format indicating how the predicate is used. A special notation is used with each argument to the predicate. A symbol appears before each argument indicating whether or not the argument must be instantiated at the time the predicate is used. An instantiated argument is one in which the argument is bound to some value. The symbols are:

- + A plus sign in front of an argument indicates an input argument. Generally, the argument is required to be instantiated. However, in some cases, the argument may be uninstantiated, as is true if the value for the argument is returned by the system.
- A minus sign in front of an argument indicates an output argument. The predicate generally returns an instantiated value that is unified with the value given when the predicate is called. However, in some cases, the value returned is uninstantiated and will always unify with the value given when the predicate is called.
- ? A question mark in front of an argument indicates that the argument may be used either for input or output, depending on the context in which it is used. The predicate description will describe the effect of supplying both an instantiated and an uninstantiated argument.

Predicates not described below are standard predicates and are described in the Arity PROLOG Language Reference Manual. All of the predicates in the Arity PROLOG Language Reference Manual may not appear familiar to

someone versed in the Edinburgh standard because Arity has defined a number of predicates that extend Arity PROLOG beyond the Edinburgh syntax. These extensions are well documented in the manuals.

btree\_key( +VarName, -Key ) (RAPSDHF)

Provides the key for btree storage of the variable in VarName.

btree\_key\_d( +DispVal, -Key ) (RAPSDHF)

Returns the appropriate btree key for the indicated disposition value.

check\_expert (RAPS)

A conclusion recorded under \$expert indicates expert should be invoked with the conclusion as the argument. If multiple conclusions are recorded, they will cause expert to be invoked repeatedly until all are processed.

check\_quit (RAPS)

Anything recorded under \$quit indicates RAPS should stop execution.

collect\_dr( +RecType, +SubRec, -Rec ) (RAPSDHF)

Collects all the elements of a specified data record from val clauses and constructs a data record. The values stored in val clauses that are not of the indicated type are asserted as temp clauses. After all values have been placed into the record, collect\_dr invokes temp\_to\_val to move the temps back to vals.

conc( +List1, +List2, -NewList ) (RAPSKBI)

Concatenates List1 and List2 to form NewList.

**dba( +VarName -Val ) (RAPSKBI)**

Returns an atom from any RAPS database for use by the RAPS Inference Engine.

**dbt( +VarName, -Val ) (RAPSKBI)**

Returns a string from any RAPS database for use by the RAPS Inference Engine.

**dbv( +VarName, -Val ) (RAPSKBI)**

Returns a value from any RAPS database for use by the RAPS Inference Engine.

**dccv( +DialogName, +/-CtrlN, +/-CtrlType, +/-VarName ) (RAPSDCV)**

Represents a table indicating, for each dialog, the variable type, name, and default value at specific control numbers. This table allows RAPS to perform several processes with regard to dialogs. First, dccv allows the program to assign default values to fields when the dialog is displayed. If there is a value for the field, then it is displayed; otherwise, the specified default is displayed. Second, by providing the type of dialog control, dccv allows RAPS to process the dialog controls in a general fashion, without the rest of the RAPS code knowing specifically which dialog is being processed. (This eliminates the need for explicit code for processing each dialog separately, which would have unnecessarily lengthened and complicated the code and code maintenance.)

**default( +VarName, -DefValue ) (RAPSDCV)**

Allows definition of the default value for variables. This predicate also allows default values to be defined in terms of existing values of other variables.

**dh\_helpDialog( +Message, +DialogName ) (RAPSDHF)**

Processes dialog Help requests for context-sensitive help (help applicable to the current dialog being processed).

Help is initiated on the next\_ctrl(0,1) message because if it is done on init\_dialog, where it might be more appropriate, then the cursor is repositioned when next\_ctrl(0,1) is processed by the default function. First, the Help key is created by using the recorded menu message to look up (using mnuatr) the menu name, which is then prefixed by an asterisk. This results in a menu message such as "procRelat" being transformed into "\*Processing/Relat." Such a transformed key is used to search the Help lines recorded under \$dialog\_help to find which line of the Help file should be initially displayed. If no match is made, the resultant failure results in help being displayed at the beginning of the file. If the key is located, a set position message is sent to the edit box control followed by an update message. Finally, the message is passed to the default dialog message handler.

All messages other than next\_ctrl(0,1) are passed to the default dialog function.

At present, help is available only at the dialog box level, not the level of a control within a dialog box. This predicate is, however, easily extendable to provide control-level help. But to do so, it is necessary to increase the size of the dialog Help file beyond 60K. A solution would be to compress the Help file and expand it as required for control-level help.

dh\_guf( +DlgMessage, +DlgName )

(RAPSDHF)

This predicate is defined as a generalized dialog message handler. Because RAPS has over 25 dialogs, it would have been prohibitive to define individual handlers for each of the operations associated with each of the dialogs. The compromise (which is considered a bit slower in execution) was to define a generalized predicate from the message perspective. That is, there is an occurrence of dh\_guf that processes the init\_dialog message, regardless of which dialog sends the message. In a few instances, the predicate definition became a bit complex because a few dialogs needed unique processing. The individual messages processed are defined as follows:

init\_dialog

Places existing variable values or default values in dialog fields before the dialog is displayed. Two different init\_dialog options are defined. The first is to process recRegister and leadAdd,

which are dialogs used to enter new recruiter or new lead occurrence information. These two dialogs are distinct because default information will always be displayed in the fields of these dialogs. The second definition handles all other initializations. If there is a current value, it is placed in the field. If no current value exists, then the defined default value is displayed. Both initial definitions ultimately fail and the `init_dialog` message is finally passed to the default dialog function for final processing.

Upon exit, all dialog boxes return a command message that indicates which action should be taken. The following command messages can be processed by `dh_guf`.

`command( _, add )`

Adds data entered in the dialog box to a data record.

`command( _, addmult )`

Adds data entered in the dialog box to a list element in a data record. A list element is one in which there may be multiple occurrences (e.g., multiple schools, multiple law violations, etc.).

`command( _, delete )`

Deletes data entered in the dialog box from a data record.

`command( _ delmult )`

Deletes data entered in the dialog box from a list element in a data record. List elements are those in which there may be multiple occurrences (e.g., multiple schools, multiple law violations, etc.).

`command( _, nxtnmult )`

Advances to the next data element in a list of data elements in the data record. That is, if a list of law violations occurs, advance to the data relevant to the next law violation.



`command( _, print )`

Prints the data element specified in the dialog box.

`command( _, prvmult )`

Backs up to the previous data element in a list of data elements in the data record. For example, if a list of law violations exists, back up to the data relevant to the previous law violation.

`command( _, quit )`

Quits from RAPS.

`command( _, select )`

Collects the data from the record associated with the key name selected in the select dialog box.

`command( _, update )`

Modifies variable values in the data record. For single-valued variables update also implies addition of the first value to the variable. Update is accommodated in three steps: (1) get the dialog controls and move them to the variables; (2) for the prosDisposition dialog, make sure the phase change is accommodated; and (3) check the menu option and store a message that the update was performed.

`command( _, updmult )`

Modifies variable values in a list element of the data record.

`char( A, 15 )`

Allows the Tab character to escape from an edit box. The predicate examines every Tab character returned by a dialog box. It then checks to see which control is being accessed, and whether that control is an edit box. If it is an edit box, a Control-Q is passed to the Arity dialog control and then the Tab or Shift+Tab is passed.

char( 0, 59 )

Examines all characters returned from dialog boxes to see if they are the F1 key for accessing help.

default

Passes all commands and characters not explicitly trapped by the dialog handler general utility function, dh\_guf, to the Arity dialog control for normal processing to resume.

dispmnu( +LDisp, -MnuNam ) (RAPSDHS)

Determines the menu associated with a lead disposition. If a lead has a particular disposition then processing has progressed to the corresponding phase.

do\_key( +Key ) (RAPS)

Executes the user key selection. The options provided are F1 for help; Shift+F10 for Quit; Alt to access the Menu Bar; and Alt+Scan Code to access a specific menu in Pull-Down mode. do\_key accomplishes its function by invoking do\_mnu with the proper argument.

dispmnu( +LDisp, -MnuNam ) (RAPSDHS)

When provided with the lead disposition (LDisp), dispmnu returns the menu name associated with that disposition (MnuNam). dispmnu is used in conjunction with phase transitions when the applicant being processed changes disposition.

do\_mnu( +MenuSelection ) (RAPS)

Performs initial processing of a menu selection message. The message is the name of the option selected, and in most instances (except for Shell, Quit, etc ), the message is also the name of the dialog that is to be displayed. First, menu\_msg is used to extract the message from a possible check/uncheck message. The message is then recorded under

\$menu for possible later use by RAPS. mnuatr is invoked to identify the Scan Code (Accelerator Key) for the selected menu option. This allows the appropriate key to be recorded under \$key to ensure the same Pull-Down Menu is displayed after the current message is processed. do\_mnu1 is then invoked to process the message.

The actions of do\_mnu are summarized here:

1. If the message is a cancel generated by the user pressing the Esc key then nothing is done. A return is processed, and the user exits from the menu mode back to the Main Window.
2. Because the menu message may be encapsulated in another message such as checked or unchecked, menu\_msg is invoked to strip off the encapsulating part, if it exists.
3. The menu message is recorded under the global variable \$menu so other RAPS predicates may access the initial menu message that initiated the current execution sequence.
4. To be able to return to the same Pull-Down Menu once a dialog is completed an Alt (Scan) message is stored under the \$key global so the beginning of the program will find it just as if the user had entered the Alt+ScanKey. This action is accomplished by passing the menu message to mnuatr which returns the Scan code for that message. To accomplish this process, menu messages must be unique.
5. Finally, do\_menu invokes do\_menu1 to process the message.

do\_mnu1( +MenuSelection )

(RAPS)

Defines any unique processing that must be done for the various possible menu selection messages. In most instances, the action is to invoke a dialog that is of the same name as the menu message. When this is not the case, the exception action is explicitly described for the message. All menu messages were designed to be unique, yet for several stages in RAPS the same dialog must be accessed. To accomplish this, certain do\_mnu1 definitions are nothing more than translations from the menu message to the appropriate dialog name.

As a default, when there is no special processing, the menu message is simply passed to dialog\_run for the appropriate dialog to be displayed.

drv( +DialogName, +CtrlNum, +VarName, ?VarValue ) (RAPSDCV)

Provides a translation from a dialog name and control number and variable name to the displayed radio button value. This allows storage of the radio button value in the record and then subsequently allows the appropriate radio button to be turned back on when the record is selected for redisplay.

dump\_key( +Key ) (RAPSDHS)

Writes all information recorded under the specified key. dump\_key accomplishes the write in a line oriented format so each recorded instance is followed by a new line.

erase\_one( +Key ) (RAPSUTL)

Erases an item recorded under the specified key, if there is one. It is assumed that there is only one value recorded under the specified key and erase\_one always succeeds.

erecord\_list( +Key, +List ) (RAPSUTL)

Erases all items stored under the specified key, if there are any. Subsequently, erecord\_list breaks down the list and stores the individual elements as separate occurrences under the key.

expand\_dr( +Dr ) (RAPSDHF)

Expand the data record (expand\_dr) takes the values from a data record and stores them in the appropriate variable names.

expert( +Goal )

(RAPSKBI)

Invokes the RAPS Inference Engine to attempt to prove Goal.

flag\_expert

(RAPSDHS)

Records the current menu message under \$expert to be passed to expert() for conclusion at the main loop level. Because record\_one is used to do this, there can be only one flag passed back to be concluded on each pass.

get\_ctrl( +Key, +CtrlNum, +CtrlType, +VarName, VarValue ) (RAPSDHF)

Accesses the values in dialog box controls for storage in variable names and vice-versa. get\_ctrl returns the value of the specified control variable depending on the type of control.

When CtrlType is "rb" (radio button), get\_ctrl returns the text associated with the checked radio button. The text is provided by the drv definitions. If the radio button is not checked, get\_ctrl fails.

When CtrlType is "ef" (edit field), get\_ctrl returns the current text in the field.

When CtrlType is "eb" (edit box), get\_ctrl returns a list, each element being one instance of the text that was in the edit box.

When CtrlType is "lbr" (list box, radio), get\_ctrl returns the selected element of the list box.

get\_idx( +VarName, +VarVal, -Idx )

(RAPSDHF)

Looks through the list of values stored under VarName and returns the index of the variable value. If the value does not exist in the list then get\_idx returns

`get_key( -Key )`

(RAPS)

Returns the key indicating the next function to be performed. In most instances, this key will be from the user; but in certain instances RAPS needs to continue a mode (like returning to the same Pull-Down Menu after a dialog) or return to a mode after escaping to MS-DOS. This is accomplished by first looking to see if a value is recorded under \$key, and if so, it is returned; if not, then the user key is returned. The key is returned in a special format by invoking `key_val`. `keyb` returns both the ascii representation of the key and a scan code. `key_val` is used to merge these two codes into a single key representation returned by `get_key`. Essentially, if an Alt sequence is entered `key_val` returns the key in the form Alt (Scan Code) which may be used as a selection argument elsewhere.

`init_ctrl( +DlgName, +CtrlNum, +CtrlType, +VarName, +VarValue )`

(RAPSDHF)

Initializes dialog controls. There are separate instances of `init_ctrl` for each type of control used within the RAPS dialogs (i.e., radio button, edit field, edit box, and list box). `CtrlType` is defined as part of the invocation of `init_ctrl` and thus ensures that the appropriate instance of `init_ctrl` is processed.

When `CtrlType` is "rb" (radio button) `VarValue` is checked to ensure that the specified `CtrlNum` matches with the `drv` setting. If it matches, then the radio button is checked. Otherwise, the radio button is set to an unchecked state.

When `CtrlType` is "ef" (edit field) a control message is sent to set the control to `VarValue`

When `CtrlType` is "eb" (edit box) `VarValue` is expected to be a list with each element corresponding to one line of text in the edit box. In this case the list is recorded under the variable name so the control manager can display it.

When the `CtrlType` is "lbc" (list box checked), a list box is used to display lists of items from `btrees`. `btree_key` retrieves the correct key for the dialog name, then each name is retrieved and added to the control.

When the CtrlType is "lbr" (list box radio) the action is very similar to that when CtrlType is "lbc" but in this instance, only one item may be checked. This procedure is typically used for select dialogs. The first action is to invoke the lbc init\_ctrl and have the set of values displayed. Second, get\_idx is used to find where VarValue is in the list so the index may be appropriately set.

When the CtrlType is "lbp" (list box predefined radio) the action is similar to that when CtrlType is "lbr" except that the list of items that appears in the list box is predefined and does not change. In this case get\_idx is used to determine the index represented by VarVal and the index is set prior to display.

interp (RAPS)

Loads support files for test execution under the Interpreter. The files RAPSDHF.ARI, RAPSDCV.ARI, RAPSRS.D.ARI, RAPSUTL.ARI, and RAPSKBR.ARI are reconsulted into ARI.IDB first. These are the files normally compiled into RAPS.EXE and RAPS.IDB. Following these reconsults, load is invoked to load the run time files.

key\_val( +Ascii, +Scan, -Formatted ) (RAPS)

Takes the Ascii value and Scan code of the key pressed and converts it to the special formatted value to be returned. This combination allows the program to work with what appears to be a single value for a key value rather than two values.

list\_true\_facts( +List\_Facts, +Num, -List\_Ans ) (RAPSKBI)

Looks for rules which can be used to prove the current premise. If one is found, then its list of premises is disassembled. Each premise in the found rule is examined consecutively as a goal to look for additional rules which can be used to prove the current premise. This sequence is continued until there are no more rules relevant to any premise.

load

(RAPS)

Loads the support files that can not be compiled. RAPSMNU.ARI, RAPSDLG.ARI, RAPSDDC.ARI, RAPSDLG.HLP, RAPSMNU.HLP and RAPSKBR.ARI are loaded into the database (IDB) by load.

main

(RAPS)

The main entry point for RAPS; main performs initial set-up and initialization. First, a check is made to see if the dialogs have been saved in the database (IDB); if not, they are loaded along with the Help files text. A save is then preformed. Main then resizes the Main Window and defines the top and bottom windows. Appropriate messages are displayed in all windows. The final action is the invocation of main\_loop, which repeats until RAPS is terminated by the user.

main\_loop

(RAPS)

A forever loop that continues until the user indicates a desire to quit. main\_loop first invokes prtMsg to see if there are any messages to be displayed. Then a prompt is written to indicate RAPS is waiting for user input. get\_key is used to wait for, return, and display the key pressed by the user. do\_key then passes the key pressed to the system for processing. The main portion of main\_loop is within snips so when main\_loop fails, backtracking finds repeat, which succeeds, and main\_loop begins another cycle.

member( +Item, +List )

(RAPSDHF)

Determine if a specific item occurs in a list. If the item is anywhere within the list, member succeeds, otherwise, it fails.

menu\_check( +State )

(RAPSDHF)

Examines a line on a Pull-Down Menu for its checked or unchecked status.



When State is set to "check" the predicate retrieves the recorded menu message; uses mnuatr to ensure the message represents a checkable menu item; if a State is recorded, it is retracted; then the checked state is assert to be true; and a menu message is sent to check the menu item.

When State is set to "uncheck" the predicate retrieves the recorded menu message; uses mnuatr to ensure the message represents a checkable menu item; if a State is recorded, it is retracted; then the checked state is asserted to be false; and a menu message is sent to uncheck the menu item.

When State is null; both cases fail and menu\_check succeeds in doing nothing.

menu\_check\_off (RAPSDHF)

Removes a check mark from the appropriate line of a Pull-Down Menu to indicate that the feature has not been accessed.

menu\_check\_on (RAPSDHF)

Places a check mark on the appropriate line of a Pull-Down Menu to indicate that the feature has been accessed.

menu\_msg( +Selection, -Message ) (RAPS)

Extracts the menu message when it is embedded in a checked/unchecked form. For those menu options that are allowed to have a status indicated with a check, Arity returns the menu status with the message by returning the message in the form checked(message) or unchecked(message). menu\_msg extracts the message if it is so embedded.

menu\_off( +Level ) (RAPSDHF)

Turns a menu items off (greys it).

menu\_on( +MenuName) (RAPSDHF)

Turns a menu item on (ungreys it).

menu\_set( +MenuName, MenuMessage, +Level, +State ) (RAPSDHF)

Sets a selected set of menus grey or ungrey. menu\_set works with either a specific set of menus or a specific menu message.

mnuatr( +MName, +MMsg, -ScanCode, -CanChk, -CanGrey, -DlgHelpKey)  
(RAPSUTL)

Defines attributes of and data associated with the Pull-Down Menu options. The various characteristics are described as follows:

MName: a code representing the menu bar label for the menu.

MMsg: the menu return message when the menu option is selected.

ScanCode: the accelerator key for the menu represented by MName.

CanCheck: a flag (y/n) indicating whether this option can be checked.

CanGrey: a flag (y/n) indicating whether this option can be greyed. There are two levels of grey options defined (y1 and y2) that allow for determining which are select, delete, and print options, and which ones can only be accessed after a selection is made.

DlgHelpKey: the Help file key for the menu option. This is used so the user does not see possibly ambiguous abbreviations in the Help file used for locators. (It would have been much easier to use MMsg values as tags in the Help files but this was not done because of the possible confusion.)

mult\_which( +Function, -Where, +DlgName ) (RAPSDHF)

Determines where in a record to act upon a multi-valued (list) element. When Function is "delete" or "update," the multi-value variable index is the current recorded value. When other functions are attempted, the

index is determined using the current index, and the number of arguments. mult\_which\_aux is used for other functions.

mult\_which\_aux( +Function, -Where, +Nargs ) (RAPSDHF)

Determines the value of the index (Which) of the list of multi-valued variables to be accessed.

When Function is "add" the current number of arguments is incremented by one and is returned in Which.

When Function is "nxt" Which is set to one more than the current index or the last argument, whichever is least.

When Function is "prv" then Which is set to one less than the current index or the first argument whichever is greater.

no\_rule\_concludes( +Fact ) (RAPSKBI)

Asks question in a negative sense "Is there no rule that concludes this premise?" If the answer to the question is true, then the information must be looked for in the database, or must be queried from the user.

nth\_arg( +Nth, +List, -NthArg ) (RAPSDHS)

Returns the Nth argument of a list.

nth\_arg\_aux( +Nth, +Curr, +List, -NthArg ) (RAPSDHS)

Assists in the selection of the Nth argument of a list. This predicate steps through the list incrementing the current argument index until it is equal to Nth; now the Head of the list contains the sought argument and is returned. Otherwise, the current index is incremented and the Tail of the list is passed recursively to the predicate for reconsideration. If a null list is encountered, the null string is returned.

phase( -Phase ) (RAPSUTL)

Returns the current processing phase (Leads, Prospecting, Selling, Processing, or Maintenance).

phase1( +MenuName, -Phase ) (RAPSUTL)

Provides support for the phase predicate to return the current phase.

process( +Ans, +Question, +List\_Ans, -Ans ) (RAPSKBI)

Accepts the user's "yes", "no" or "why" response. If the response is "yes" or "no," the reply is asserted. If the response is "why," a dialog is begun with the user to illustrate the knowledge base logic.

prtMsg (RAPS)

Prints all messages recorded under the \$prtMsg key and deletes them after they are printed

question\_reply( +Question, +/-List\_Ans, -Resp ) (RAPSKBI)

Examines a question to see if it has been replied to previously. The Inference Engine is creating predicates of the form "replied(Question,Response)." When question\_reply encounters a question that has been answered before, it returns that answer. If the question is unanswered, it answers the question either from the database or through a dialog with the user.

quit (RAPSUTL)

Performs cleanup prior to exiting RAPS. First, the Main Window is made the current window and recolored, then the top and bottom windows are deleted. All values stored under keys \$key, \$prtMsg, and \$quit are erased. The screen is then cleared and the program aborted. When running in the Interpreter the user is returned to the interactive mode. When executing the compiled version, abort returns the user to MS-DOS

recMsg( +Message ) (RAPSDHF)

Records a message for display as the result of a user key press.

record\_list( +Key, +List ) (RAPSUTL)

Records the elements of the supplied list under the specified key. record\_list does not check for current values recorded under the specified key (see erecord\_list).

record\_one( +Key, +Value ) (RAPSUTL)

Records a value under the specified key. If a value is already recorded under the key, then it is replaced. This function assumes there will be only one value recorded under the specified key.

remove\_vals( +RecType ) (RAPSDHF)

Removes the record type variables from a data record.

replied( +Question, +/-List\_Ans, -Resp ) (RAPSKBI)

Created by the Inference Engine to maintain track of questions asked and answers given.

repos\_window( ( X1, Y1 ), ( X2, Y2 ) ) (RAPSUTL)

Resizes and/or moves the current window to fit the coordinates (X1,Y1) and (X2,Y2). This predicate overcomes the restriction of not being able to redefine an existing window in Arity PROLOG. An existing window that needs to be moved or changed in size is modified using repos\_window.

repos\_aux( +MoveXDir, +MoveYDir, +ResizeXDir, +ResizeYDir )  
(RAPSUTL)

A support predicate used by repos\_window.

rule( +Type, +Fact, +Seq, +Premise ) (RAPSKBR)

Defines the rule structure used in the knowledge base. This structure is not accessible to the user, who must use the knowledge base interface to access and edit rules.

rv( +RecType, +/-VarName, -VarType ) (RAPSRSD)

Define the record type for a variable and the variable type of the item.

save\_dr( +VarName ) (RAPSDHF)

Saves the specified data record. Saving requires that all variables be collected into a single list and stored under the btree key provided by the btree\_key function. The existing record under its initial key is removed prior to saving the current record. For data records, the recorded select\_key and select\_name are used to ensure that the original record is removed even if the name and phase have changed since the record was selected or created.

save\_idb (RAPSUTL)

Saves the database.

temp\_to\_val (RAPSDHF)

Reasserts the val items stored in temp.

today( -Today ) (RAPSUTL)

Gets the system date in the form date( Y, M, D ), and returns it in the form M/D/Y where Y is the two digit year ( 89 for 1989 ).

`true_fact( +Fact, -List_Ans )`

(RAPSKBI)

Allows the Inference Engine to walk through a series of related rules. expert is called with a goal. `true_fact` looks to see if rules exist in which that goal is a premise. Then these new premises become new goals for expert.

`trule( +Type, +Fact, +Seq, +Premise )`

(RAPSDHF)

Temporary rules which are user created. `trules` are reconsulted and converted to rules. Users are not allowed to edit rules, because reconsulting a rule will remove all occurrences of a rule predicate.

`update_bottom`

(RAPSUTL)

Prints messages on the bottom of the screen referring to recruiter, candidate, and phase.

`val( +VarName, -VarValue )`

(RAPSDHF)

Returns the value of the specified variable, which will be a list. `val` is defined as external in RAPSDHF but is not coded within the module. `val` is asserted at run time and thus resides in the database rather than in the compiled code. There exists an occurrence of `val` for every variable for which a value has been entered.

`val_mod( +VarName, +Function, +Where, +ModValue )`

(RAPSDHF)

Modifies the data item value in `VarName`. The mechanism by which `val_mod` works depends on the function. With multi-valued variables (lists of lists) the assumption is that there is a current value for the variable instantiated as `val(VarName, VarValue)`. When Function is "add" a new value is appended to the end of the list. When Function is "del" the Nth item of the list is deleted. When Function is "upd" the Nth item of the list is replaced.

`val_mod_add( +CurrList, +AddValue, -NewList )` (RAPSDHS)

Appends AddValue to the end of the CurrList producing NewList

`val_mod_aux( +VarName, -CurrValue )` (RAPSDHS)

If a value for the variable exists, `val_mod_aux` retracts the instance `val` and returns the current value. If there is no current value, `val_mod_aux` returns a null list.

`val_mod_del( +CurrList, +CurrPtr, +DelPtr, -UpdList )` (RAPSDHS)

Deletes the DelPtr value in CurrList. This is done by transferring CurrList elements to UpdList and when CurrPtr is equal to DelPtr nothing is transferred from CurrList to UpdList.

`val_mod_upd(+CurrList,+CurrPtr,+UpdPtr,+UpdVal,-UpdList)`  
(RAPSDHS)

Updates the UpdPtr value in CurrList with UpdVal. This is done by transferring CurrList elements to UpdList and when CurrPtr is equal to UpdPtr then substitute UpdVal for the element to be moved from CurrList to UpdList.

`val_set( +VarName, +NewValue )` (RAPSDHS)

Sets the value of the variable in var as appropriate.



## 8. RAPS Printer Interface

The RAPS printer interface is written to be maximally flexible. A major concern was the realization that forms and USAREC documents are subject to continual change and revision. Consequently, it was considered desirable to allow RAPS to adapt to changing USAREC, DA, and DOD forms.

At some point in the future, it is likely that forms will be maintained only in electronic form. In electronic form, they can be most easily adapted, revised, and reproduced. And, there are many commercial tools available to copy printed forms, and to maintain and edit electronic forms (cf. FormWorx\* ).

RAPS was designed to allow these electronically maintained images to be transferred to the printer along with simple predicates that drop data from RAPS records into the form at the proper coordinates. These predicates are of the form

```
printVar(Xposition,Yposition,VariableName).
```

To use this predicate, one merely inserts the horizontal position (Xposition) measured in inches from the left hand side of the form; the vertical position (Yposition) measured in inches from the top of the form; and the name of the variable in the database that should be printed at that position (VariableName). This simple predicate positions the data on any form.

Forms that are maintained electronically (either with FormWorx, FormTool, Word Perfect, or any other means) may be printed by passing the binary image of the electronic form to the printer. This process is equivalent to the DOS command

```
D:>COPY /B ELECTRON.FRM LPT1: .
```

If a form can be printed in this fashion, then RAPS can use the same binary image of the form in its printFile(X) predicate:

```
printFile('electron.frm').
```

---

\* FormWorx Corporation  
1601 Trapelo Road  
Waltham, MA 02154

The mechanisms described above provide great flexibility. New forms can be defined at any time through the use of any electronic form building tool. Any data maintained in the RAPS database can be output on any form.

All of the RAPS databases (lead database, prospect database, applicant database, and DEP/DTP database) are accessible. Consequently, not only can the DD-1966 and standard DA forms be maintained, but RAPS can be used to send form letters to police agencies, schools, employers, etc. These capabilities represent a considerable extension and should significantly improve user acceptance of the automated system.

### **8.1. RAPS Printer Predicates**

Below are described the various RAPS printer predicates that may be used for forms generation. All printer predicates are in the file RAPSUTL.ARI.

`fileClose(FileName)` (RAPSUTL)

Closes FileName.

`fileOpen(FileName)` (RAPSUTL)

Opens FileName for printing.

`filePrint(FileName)` (RAPSUTL)

Prints FileName to the printer.

`printAt(Xinches,Yinches)` (RAPSUTL)

Positions the print position at coordinates Xinches and Yinches.

`printAt(Xinches,Yinches,TextString)` (RAPSUTL)

Begins printing TextString at position Xinches and Y inches.

`printBox(Xinches,Yinches,LengthInches,HeightInches)` (RAPSUTL)

Prints a box LengthInches long and HeightInches high at position Xinches, Yinches.

`printCharHeight(CharHeight)` (RAPSUTL)

Resizes the current font to be CharHeight in point.

`printClose` (RAPSUTL)

Closes the printer to output.

`printEject` (RAPSUTL)

Ejects a page on the printer.

`printForm(1966)` (RAPSUTL)

Prints the 1966 form and drops a few sample data items onto the blank form. This predicate requires that a LaserJet compatible printer is equipped with a line drawing font prior to calling this predicate. If a line drawing font cartridge is unavailable, the soft font file "LINEPRNT.FNT" can be downloaded to the printer with the command

COPY/B LINEPRNT.FNT LPT1: .

`printHoriz(LengthInches)` (RAPSUTL)

Prints a horizontal line LengthInches long.

`printList(Xinches,Yinches,List)` (RAPSUTL)

Prints a list beginning at position Xinches, Yinches. Each element of the list is printed on a new line, but at the same Xinches margin.

`printOpen` (RAPSUTL)

Opens the printer for output.

`printOrientation(Orientation)` (RAPSUTL)

Determines whether page orientation is "portrait" or "landscape".

`printPitch(Pitch)` (RAPSUTL)

Assigns pitch Pitch to the current font.

`printSpacing(Spacing)` (RAPSUTL)

Determines whether spacing is "fixed" or "proportional".

`printString(TextString)` (RAPSUTL)

Prints TextString at the current print position.

`printStroke(Stroke)` (RAPSUTL)

Assigns Stroke to the current font weight. Stroke can be either "bold", "normal", or "light".

`printStyle(Style)` (RAPSUTL)

Assigns Style to the current font. Style can have a value of "upright" or "italic".

`printSymbol(Symbol)` (RAPSUTL)

Selects the Symbol set for the current font. Symbol defaults to "roman8".

`printTop` (RAPSUTL)

Places the print position at the top left hand corner of the page.

`printTypeface(Typeface)` (RAPSUTL)

Assigns Typeface to the current font. Typeface can have the following values: "courier", "helv", "tmsRmn".

`printVar(Xinches,Yinches,DatabaseVariable)` (RAPSUTL)

Prints DatabaseVariable at print position Xinches, Yinches.

`printVert(LengthInches)` (RAPSUTL)

Prints a vertical line of LengthInches at the current print position.

## **9. CONCLUSION**

RAPS represents a proof-of-concept in the development of a user-friendly expert system designed to be a recruiter's assistant. In many ways, RAPS exceeds the modest research study objectives, which were to develop an expert system prototype.

In our development efforts, the usual conflicts between breadth and depth were resolved in favor of breadth. RAPS is designed as a full-featured system. In addition to codifying the 601-210 regulations and functioning as a Recruit Qualifier, RAPS includes provisions for a Form Builder. The Form Builder will fill out forms for completing waivers, for requesting police records, for requesting copies of school records, etc. By enabling multiple functions, including simplifying the maintenance of LRLs, REACT files, 200 cards, etc., RAPS is more likely to be viewed as a valuable recruiter's assistant instead of as another instance of recruiter "micro-management" by USAREC.

### **9.1. The RAPS Recruit Qualifier**

The Recruit Qualifier is embodied in the RAPS knowledge base and Inference Engine. The Recruit Qualifier can be used to summarize the regulations governing eligibility criteria, policies, and procedures for enlistment and processing of applicants into the Regular Army and the U.S. Army Reserve as summarized in Army Regulation 601-210 and various USAREC regulations. The Recruit Qualifier is intended to help recruiters adjust the processing and documentation of applicants in accordance with their varying abilities and personal characteristics.

The Recruit Qualifier is capable of continually reviewing each recruiter's leads and applicants and their current state of processing so as to guide the recruiter in the subsequent actions necessary to qualify the recruits. This is done automatically through a review of the entered data by the Inference Engine, and the display of questions and information to the user. One component of this process is evident in the RAPS menu system which greys the processing actions which are irrelevant to the applicant, and checks the completed actions. The grey/ungrey and checked/unchecked states can be under the control of the Inference Engine. In addition, the Recruit Qualifier is built with the capability to automate the completion of waiver requests, guide the submission of waiver requests through channels, and help the

recruiter budget his/her time so that unlikely waiver requests do not needlessly occupy his time.

The Recruit Qualifier is built on the foundation of the knowledge base of rules which correspond to USAREC and Army regulations. Because regulations are not static, but tend to change on a regular basis, these changes can be incorporated in the knowledge base rules and the expert system Inference Engine will behave in accordance with the current rule set. Maintenance of the knowledge base is allowed to provide RAPS with maximum flexibility in accommodating changes in regulations.

## **9.2. The RAPS Form Builder**

The RAPS Form Builder is embodied in the printer interface. The printer can be accessed from any point in a RAPS session, and can be automatically accessed by the RAPS Inference Engine to generate forms, waiver requests, etc. The integral printer interface allows a recruiter to complete his or her paperwork interactively through RAPS. The expert system Inference Engine can be loaded with sufficient intelligence to guide the recruiter in collecting all of the data necessary to complete valid DOD and DA forms. The data records can then be accessed by the Form Builder to output forms on laser printers, or to complete pre-printed forms on standard dot-matrix or impact printers.

## **9.3. RAPS Tuning and Extensions**

The development team working on RAPS had access to excellent briefings with and input from USAREC. There is, however, no substitute for user experience with a system under development. RAPS can benefit from such input for tuning the user interface, as well as for tuning dialog boxes.

### **9.3.1. RAPS User Interface**

The system of RAPS menus and dialog boxes was abstracted from current USAREC manuals and procedures. In many cases, existing USAREC forms were simply copied into electronic forms. It is important to review the logical organization of RAPS and the dialog sequences to assure their relevance and ease of use. At this point, we estimate that approximately 25% of the existing dialog boxes could be eliminated. This reduction in

complexity could be accomplished by merging similar dialogs and eliminating the few duplications that exist. Such tuning would have a corresponding effect on the menu interface and result in a much simpler interface.

RAPS can benefit from the addition of point-and-shoot support from mouse or light pen hardware.

#### **9.3.1.1. Dialog Box Tuning**

Because RAPS functions through dialog boxes, it is important that the form and function of these boxes are efficiently and logically organized. User experience will indicate whether or not the defined hot-keys are logical and useful. Similarly, user experience can determine proper defaults. Field descriptions may need explanation, clarification, editing, or removal. Likewise, some extensions to the RAPS editor may be appropriate (for example, to word wrap within edit boxes).

RAPS is delivered with source code and with tools that can be used to modify the appearance and function of dialog boxes. User experience with RAPS can guide in tuning RAPS to be logically consistent with the work habits of recruiters.

#### **9.3.1.2. Inference Engine Access and Control**

The RAPS Inference Engine is entered with a single predicate, "expert(\_)." expert(\_) can be used to prove any goal in the parenthesis. Consequently, the Inference Engine can be invoked from any RAPS menu item, from any dialog box, or from any field within any dialog box. For example, the Inference Engine can be invoked at the menu level to prove whether or not a lead qualifies as a prospect. The Inference Engine can be also invoked at the dialog box level to assure that all information relevant to the 714 form is complete and ready for transmission to ARADS. Similarly, the Inference Engine can be invoked at the field level to prove that a telephone number or address has been entered in the correct format.

Currently, expert(\_) is invoked from "update" or "add" functions upon exit from a dialog box, at the point of data collection for record construction, and at the point of data parsing when RAPS accesses a record. Each time expert(\_) is invoked, the Inference Engine looks to see what goal or goals



should be processed at the time. The Inference Engine then systematically accesses each relevant rule and attempts to prove the goal(s).

How to best utilize the Inference Engine from each field, from each dialog box, and from data reads and data writes requires additional experience as well as the development of corresponding rule sets.

### **9.3.2. RAPS Extensions**

The developers began this project with a system design document that was believed to be completely comprehensive and as extensive as necessary. In implementing the system design document, however, the dynamic nature of USAREC operations quickly became apparent. During our short development time, new Army regulations on recruiting were published twice. Moreover, USAREC decided to codify and maintain physical test data on applicants. Forms have been revised, but data communications difficulties with ARADS have not yet been completely resolved.

Due to the dynamic nature of recruiting as described above, as well as the developer's widening perspective of recruiting requirements, RAPS development concentrated on some areas that were not anticipated in the original system design document. As a result, many more hooks were left open in the system to accommodate future extensions and possible implementations.

#### **9.3.2.1. Forms Printing**

All of the RAPS databases (lead database, prospect database, applicant database, and DEP/DTP database) are built to be accessible to the Form Builder. Consequently, in addition to having hooks for print drivers to complete DD-1966 and standard DA forms, RAPS has an extensive set of predicates which allow any database information to be printed at any location on any form. Additionally, RAPS is built with the capability to print any form definition which can be maintained on disk as a string of printer commands. These capabilities can be used to build form letters to police agencies, schools, employers, etc. and to modify or add any new forms to the RAPS system. These capabilities represent a considerable extension and should significantly improve user acceptance of the automated system.

#### **9.3.2.2. Field Format Validation**

Two methods are available to validate field entries in dialog boxes. The first involves writing computer code specific to the field. The second involves invoking the Inference Engine. Both options remain open; experimentation will suggest which option is most appropriate for each field.

#### **9.3.2.3. Data Communications**

RAPS employs a number of predicates to assemble and disassemble data records to ASCII strings. These predicates can be harnessed to allow data and modem communications with ARADS and with telephone modems to allow RAPS to automatically assist recruiters with telephone prospecting. When ARADS is equipped with a data-transfer protocol, the corresponding driver may be hooked to RAPS to allow error-free data transfer.

#### **9.3.2.4. Record Length Optimization**

RAPS already employs many state-of-the art methods for minimizing database size. RAPS does not store default values in the database. RAPS employs keyed, variable length records. Record lengths may be minimized, however, by examining other fields for common responses and again not storing the more common (default) responses. With improvements in default management, record lengths can be reduced. A further reduction in record lengths was accomplished by employing two-character codes to identify field variables in a record, instead of lengthy variable names.

#### **9.3.2.5. Help Files**

Currently RAPS provides help at the dialog box level using 71 kilobytes of space in the RAPS database. Provisions have been made to provide help at the field level, however, this would require a significant increase in the size of the help files beyond the current 71 kilobytes.

To provide context sensitive help at the field level and to improve the speed of response to a help request, it may be desirable to compress the help files using any of the available new file compression techniques and to decrypt the file as it is displayed. Simultaneously, the help file might be keyed in the database instead of sequentially searched for proper positioning.

### **9.3.3. Additional RAPS Features**

Two features have been discussed that can significantly improve user acceptance of RAPS. The first feature is a pop-up MANDEX calendar; the second is a command interface.

RAPS can be significantly improved by the addition of a pop-up MANDEX that can be maintained by both the user and RAPS. This calendar could be used to maintain milestone dates for each lead, prospect, applicant, and recruit managed by a recruiter. Automated contacts (e.g., telephone calls placed through RAPS) could be logged to this calendar for command review. Likewise, sales interviews and prospect contacts that are initiated through RAPS could be automatically logged to the calendar. Non-RAPS initiated activities could be entered manually by the user.

RAPS could also benefit from a command interface that allows the station commander to have read-only access to each recruiter's database of leads, prospects, applicants, and recruits. Such access could enable better management of Mission Box requirements and performance projections. Additionally, it may be useful to allow the station commander to have write-only access to each recruiter's MANDEX calendar to allow the commander to assist recruiters in scheduling and managing their time.

RAPS has been built with both of these potential features in mind. The necessary hooks to the RAPS database as well as to the RAPS interface already exist to permit such implementations.

#### **9.3.3.1. RAPS Access by Company and Battalion Leadership**

One further potential use of RAPS deserves discussion. Current Recruiting Operations, with their increasing reliance on ARADS, do not adhere to the Army chain of command. The implementation of ARADS effectively ties each recruiting station to the brigade PRIME computer in Louisville, Kentucky. Each station commander submits the Automated Data Record and projections to Louisville, bypassing the usual chain of command. Company and battalion commanders are essentially out of the loop and can retrieve data from the Louisville computer only incidentally. RAPS offers the potential to maintain chain of command accountability by the designated leadership teams.

With the implementation of data communications capabilities to RAPS, it is possible to leave the RAPS system on-line to monitor telephone lines during off-duty hours. Command versions of RAPS at the company and battalion levels could automatically poll the RAPS stations to retrieve summary statistics about leads, prospects, applicants, and recruits. Automatically, these data could be passed up the chain of command to keep company and battalion leadership teams apprised of local performance.

## **10. RECOMMENDATIONS**

Recruiters will be equipped with personal computers in the future. Hardware support for recruiters will be provided because records can be maintained more cost-effectively on personal computers. And, there is ample evidence that recruiters need support in completing their mission.

The introduction of personal computer hardware to recruiting will not satisfy any recruiter needs unless useful, user-friendly software already exists when hardware becomes available. If there is no software, unworkable hardware will discourage reliance on the machines that are introduced.

We recommend that the development work on RAPS continue in anticipation of the time when RAPS is fielded. The recommendations below outline the tasks necessary to field RAPS.

### **10.1. User-Developer Testing and Tuning**

A minimum of six weeks of user-developer concerted testing and tuning effort is required. This testing and tuning effort would begin each week with one naive user tasked with working on RAPS for an 8-hour period. The developers and the user would jointly identify potential improvements ranging from menu choices and functions, through dialog box functions, to field level improvements involving data entry, field identifiers, and defaults.

The remainder of each week would be devoted to implementing the improvements identified. The process of naive user testing would continue until the users and developers agreed on the suitability of the interface. Finally, a two week period would be used to clean up and tighten the code changes.

### **10.2. Rule Construction and Inference Engine Adjustment**

The real value of RAPS is in the knowledge base. It is important to begin by translating AR601-210 and USAREC regulations into rules for the knowledge base. This translation should begin immediately.

Simultaneously, the Inference Engine should be unharnessed and given authority to reject data, reject applicants, and deny the recruiter access to

RAPS functions based upon the quality and nature of data. Presently the Inference Engine cannot alter the RAPS program flow. The Inference Engine in its present state is used only illustratively and to demonstrate the informative capabilities of the knowledge base. With a more complete rule set, the Inference Engine can be given more authority.

### **10.3. Point-and-Shoot Support**

The RAPS interface can benefit greatly from the addition of a point-and-shoot interface that supports the use of a Mouse and/or light pen. Programming to implement mouse and light pen support can begin immediately.

### **10.4. Data and Text Compression Algorithms**

It is also important to address the size of RAPS. To implement RAPS on a dual floppy system, it will be helpful to reduce the size of the text files maintained within the raps database. Especially important is the reduction in size of the Help files.

The Lempel-Ziv-Welch\* (LZW) compression technique can be applied to the Help files to reduce their size by at least 50%. This application can also allow help to be offered at the field level within dialog boxes; and will result in significantly faster help displays.

An additional application for a compression technique is to reduce the size of form descriptions and allow these to be maintained within the RAPS database. Maintaining form descriptions in compressed form within the database simplifies distribution and software maintenance. The concern about distributing a number of related files disappears when only two files need to be distributed (RAPS.EXE and RAPS.IDB).

To minimize the memory requirements of RAPS, it may also be useful to investigate linking RAPS components with a sophisticated overlay linker like

---

\* Welch, Terry. "A Technique for High Performance Data Compression," *Computer*, June 1984.

Ziv, J. and Lempel, A. "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, May 1977.

PLINK86. This linker is reputedly able to significantly reduce the size of compiled applications.

Examination of compression techniques can begin immediately.

## **10.5 Forms Generation and Development**

The RAPS printer interface employs a flexible mechanism by which forms can be designed and maintained outside of RAPS and incorporated into RAPS at any point in time. Consequently, form generation and development can be initiated now to design and maintain those forms that RAPS should be able to complete.

Any of the available commercial tools that allows the string of printer commands to be written to disk is appropriate for designing and maintaining forms for RAPS. This work can begin immediately.

Simultaneously, the printer predicates should be improved so they can recognize the presence or absence of a printer.

## **10.6. MANDEX Implementation**

Our previous work with USAREC suggests that one major concern of recruiters is the requirement to document their daily activities. As recruiter work is implemented through RAPS, a log of recruiter phone calls, house calls, interviews, can be automatically developed and maintained in chronological order. Likewise, an additional function can be developed that would allow the recruiter direct access to the log to include non-RAPS activities.

These calendar activities represent stand-alone code modules upon which implementation can begin immediately.

## **10.7. Communications module**

At present RAPS does not have any communications capabilities short of being able to import electronic files and the ability to export data to paper. A significant improvement involves the capability to send and receive data by modem, or by direct computer-to-computer connection.

RAPS will require a sophisticated error-checking communications protocol. Previous experience suggests that USAREC telephone lines are unusually noisy and that accurate data transmission requires error-checking. Any of the standard protocols can be implemented (Xmodem, CrossTalk, Kermit, Compuserve, etc.). While the choice of protocols should be made with reference to ARADS capabilities, a communications capability should be developed for RAPS.

#### **10.8. Review of USAREC procedures, operations, and data requirements**

The system of RAPS menus and dialog boxes was abstracted from current USAREC manuals and procedures. In many cases, existing USAREC forms were simply copied into electronic forms. Additionally, RAPS copies the flow sequence taught at Fort Benjamin Harrison; that recruiters work through the sequence of lead generation, prospecting, selling, and maintenance. Consequently, RAPS represents a formal codification of existing practice.

It may be appropriate to review and reconsider the USAREC procedures, operations, and data requirements to evaluate how these contribute to (1) the mission, (2) recruiter performance, (3) recruiter improprieties and (4) recruiter morale and satisfaction.

Procedures, operations and data requirements that clearly contribute to mission accomplishment, performance, ethics, and morale should be maintained. Those aspects of recruiting that do not contribute positively should be reconsidered. This re-evaluation is important to assure that RAPS not codify non-essential requirements and that RAPS accurately represents successful recruiting practices.

#### **10.9 RAPS Deployment**

We regard RAPS as having considerable potential for improving the productivity and morale of recruiters. RAPS is a tool that can significantly reduce the workload of recruiters by systemetizing the manner in which work is accomplished, by reducing errors, and by providing command staff with accurate information regarding recruiter activities.



RAPS also has the potential to interface with ARADS in at least two ways. When data communications problems with ARADS are solved, RAPS can be used to transfer applicant data records to ARADS. In the short term, however, applicant records may be transferred to MEPS on floppy disks and accessed on one MEPS terminal running DOS.

The final recommended step in RAPS development involves the limited distribution to a few well-equipped recruiting stations. Simultaneous use and tuning by the RAPS developers can transform RAPS from a working prototype to the full-featured recruiter's assistant envisioned.

This full-featured recruiter's assistant represents a solution to a number of well-perceived problems. Because RAPS codifies USAREC and Army regulations concerning recruiting, deployment of RAPS will tend to minimize recruiter errors related to learning new regulations and keeping up to date with changes in old regulations. The RAPS knowledge base is constantly consulted with changes and entries to applicant records, thus the quality of enlistment packets should improve markedly. RAPS usage ensures that USAREC and DA policies will be applied consistently across similar situations. RAPS will make the experience of policy experts available to even the most inexperienced recruiters, thereby improving recruiting station efficiency by enabling problems and questions to be resolved rapidly by any recruiter. Finally, RAPS consistent application of USAREC and DA regulations will minimize bottlenecks caused by uneven paper flow, inconsistent work habits, and "sandbagging" of applicants.

## **Appendix A - Glossary**

**Age** - Current age of applicant as of data entry date; should be computed when needed from birth date and current date. If birth date is not known RAPS, will request it when needed.

**Arity** - The number of arguments to a PROLOG predicate.

**Arrival** - Date applicant is expected to arrive at MEPS station for testing.

**Atom** - Atoms are PROLOG data structures that are textual constants. They can contain letters, digits, or symbols.

**CC Interview** - Center commander interview date. This data is required for DEP loss information dialog.

**Component** - Either RA or USAR. This item is specified as part of Form 714 testing request data.

**Court** - Generally refers to the court in which a law violation was tried. This data item appears on the law violations dialog in the Processing and Maintenance Phases of RAPS.

**Date of Entry** - A dialog box field indicating the date the data items in the dialog were entered or last updated.

**Date to Enlist** - Projected date an applicant is expected to enlist.

**Default** - The value displayed in an edit field upon initialization. Default values are chosen to minimize user editing requirements, and where possible, RAPS chooses defaults intelligently.

**DEP** - Delayed Entry Program. A program to address retention of recruits during the period between oath of enlistment and actual ship.

**DEP Date** - DEP in date (or DTP date for USAR) as indicated on automated Form 714 data.

**Development Version of RAPS** - Written in interpreted Arity PROLOG and can be examined, modified, and tested using the Arity PROLOG Interpreter.

**Disposition** - The result or outcome of a law violation; shows up on the law violations dialog in the Processing and Maintenance Phases.

**Distribution Version of RAPS** - Consists of a compiled .EXE file and an .IDB database file. These two files comprise the entire RAPS software package and can be distributed on one disk without additional licensing fees or support software.

**DTP** - Delayed Training Program. Similar to DEP but associated with the USAR as opposed to the RA.

**EIDS** - The Army's Electronic Information Delivery System, which consists of an MS-DOS-compatible microcomputer coupled with an optical laser disk that allows video sequences to be displayed on the microcomputer monitor. EIDS is the intended delivery platform for RAPS.

**Follow-Up** - Generally refers to a date in the future when contact with a lead, prospect, applicant, or recruit is to be made.

**Form 200B** - Form used to document DEP/DTP maintenance activities.

**Function Date** - Refers to the date of a function associated with DEP/DTP maintenance. The date the recruit attends the function is entered on USAREC Form 200B.

**Goal** - The solution a system attempts to reach using operators.

**Knowledge Base** - The part of an expert system containing application-specific reasoning, codified knowledge, and heuristics that the Inference Engine uses in the course of reasoning about a problem. In expert systems like RAPS, whose reasoning knowledge is represented as rules, the knowledge base is a rule set or rule base.

**Med Stat** - Flag used on automated Form 714 data record indicating whether medical results are back or not.

**Orientation Date** - The date of initial DEP/DTP orientation following oath of enlistment. This is an item of the update DEP/DTP data dialog in the RAPS Maintenance Phase.

**PADD** - Projected active duty date (or ship date for USAR).

**Place** - Shows up in the law violations dialog and refers to the location the violation took place.

**Premise** - The "IF" part of a production rule. The portion of a rule composed of one or more conditions connected by Boolean operators, such as AND, OR, XOR (exclusive OR), and NOT. If the premise can be established as being true, then the rule's conclusion is valid. A premise is an example of a logical expression.

**Proj Codes** - Codes used on Form 714 transmitted to MEPS indicating test, medical, and DEP projections.

**Projection Date** - Projection date on the current Form 714 on file.

**QT** - An abbreviation for the AFQT score; shows up most often as a dialog display label.

**Qual** - Abbreviation for qualification as a type of knowledge base rule, or the qualification portion of an applicant's name, such as Jr. or Sr.

**Qualification** - A type of rule appearing in the RAPS database. Qualification rules define the ways in which an applicant may satisfy various requirements, such as age or education, during processing.

**RC** - Two character code for recruiter data record status used on Form 714.

**Referral** - A lead referred to a recruiter by an individual during some stage of processing.

**RES** - The result code entered in the automated Form 714 data by the CG on the ARADS processing list.

**RS** - Abbreviation for recruiting station.

**Rule Item** - An individual instance of a knowledge base rule. The item may be any single instance of a qualification rule, verification rule, or a validation rule.

**Rule** - A conditional statement, in two parts, specifying a recommendation, directive, or strategy, expressed as IF premise THEN conclusion or IF condition THEN action.

**SC** - Two character code for station commander data record status on automated Form 714 data.

**Taxonomic System** - The classification system for arranging data objects in RAPS.

**T/P** - Abbreviation label used on DEP/DTP Follow-up Data dialog indicating whether the follow-up was by telephone or in person.

**Tran Date** - Date of record transmission to MEPS.

**Validation** - A type of RAPS knowledge base rule that defines the various acceptable values for an entry, such as Asian as a valid entry for a population group.

**Verification** - A type of RAPS knowledge base rule that defines the various acceptable ways in which an entry may be substantiated as true, such as using a birth certificate to verify applicant name and age.